

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO - CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN TELEMÁTICA

**DISEÑO DE APLICACIONES INTERACTIVAS T-GOVERNMENT,
T-HEALTH Y T- LEARNING PARA SU APLICACIÓN EN EL
SISTEMA DE TELEVISIÓN DIGITAL TERRESTRE DEL
ECUADOR(SBTVD) PARA LA EMPRESA TELEVISIÓN DEL
PACÍFICO S.A. GAMATV.**

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

AYALA ATI MANUEL ALEJANDRO

DIRECTOR: ING. RENATO CUMBAL

QUITO, JULIO 2011

DECLARACIÓN

Yo, Ayala Ati Manuel Alejandro, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Ayala Ati Manuel Alejandro

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Ayala Ati Manuel Alejandro bajo mi dirección.

Ing. Renato Cumbal

Director de Tesis

AGRADECIMIENTOS

A quien lea estas líneas porque eso significa que este trabajo de investigación está terminado.

Al Ingeniero Ramiro Espinosa que con su aporte y su valioso tiempo me supo guiar por el desarrollo de mi tesis.

Al Ingeniero Renato Cumbal por su apoyo en la dirección del desarrollo de mi tesis.

A Anita quien con su valiosa ayuda me permitieron finalizar esta investigación con éxito.

A la comunidad de Ginga Argentina y Ginga Brasil, su ayuda desinteresada permitieron despejar dudas y me ofrecieron la orientación en diferentes aspectos de esta investigación.

A todos quienes hacen la comunidad de software libre, sus diversas herramientas permitieron realizar este trabajo enteramente con libertad.

DEDICATORIA

A ti madre querida, tu luz me alumbra, tu voz me guía, tu mano me levanta, tu camino por el buen sendero me lleva, gracias madre mía por darme la vida, por levantarme en cada caída, por apoyarme en cada tropiezo, por alentarme en cada acierto.

De todo corazón gracias.

ÍNDICE DE CONTENIDO

CAPÍTULO 1. MARCO TEÓRICO

1.1 SÍNTESIS HISTÓRICA DE LA TELEVISIÓN.....	1
1.1.1 HISTORIA DE LA TELEVISIÓN EN EL MUNDO.....	1
1.1.1.1 La televisión mecánica.....	1
1.1.1.2 La televisión electrónica.....	1
1.1.2 HISTORIA DE LA TELEVISIÓN EN ECUADOR.....	2
1.2 LA TELEVISIÓN DIGITAL.....	5
1.2.1 TELEVISIÓN DIGITAL POR SATÉLITE.....	5
1.2.2 TELEVISIÓN DIGITAL POR CABLE.....	5
1.2.3 TELEVISIÓN DIGITAL TERRESTRE.....	6
1.2.4 IPTV.....	7
1.3 ESTÁNDARES INTERNACIONALES DE TELEVISIÓN DIGITAL TERRESTRE	8
1.3.1 ATSC (ADVANCED TELEVISION STANDARDS COMMITTEE).....	8
1.3.2 DVB-T (DIGITAL VIDEO BROADCASTING - TERRESTRIAL).....	9
1.3.3 DTMB (DIGITAL TERRESTRIAL MULTIMEDIA BROADCAST).....	10
1.3.4 ISDB-T (INTEGRATED SERVICES DIGITAL BROADCASTING - TERRESTRIAL).....	11
1.3.5 ISDB-Tb.....	11
1.3.5.1 Adopción del estándar ISDB-Tb en Ecuador.....	13
1.4 APAGÓN DE LA TELEVISIÓN ANALÓGICA EN ECUADOR.....	14
1.4.1 LA LEY DE RADIODIFUSIÓN Y TELEVISIÓN.....	15
1.4.2 LAS EMPRESAS TELEVISIVAS.....	15
1.4.3 EL CONSUMIDOR FINAL.....	15

CAPÍTULO 2. TELEVISIÓN DIGITAL INTERACTIVA

2.1 INTRODUCCIÓN.....	17
2.2 TELEVISIÓN DIGITAL.....	17
2.2.1 ESTRUCTURA GENERAL DE LA TELEVISIÓN DIGITAL.....	18
2.3 SISTEMA BRASILEÑO DE TELEVISIÓN DIGITAL.....	20
2.3.1 ARQUITECTURA DEL SBTVD.....	20

2.4 EL MIDDLEWARE GINGA.....	22
2.4.1 AMBIENTES DE PROGRAMACIÓN.....	23
2.4.1.1 Ginga-NCL.....	24
2.4.1.2 Ginga-J.....	26
2.4.1.3 Ginga-CC.....	28
2.5 USANDO LA INTERACTIVIDAD.....	30
2.5.1 RESTRICCIONES Y DIFERENCIAS CON UN COMPUTADOR.....	31
2.5.2 ESTÁNDARES ADOPTADOS PARA LA TELEVISIÓN DIGITAL TERRESTRE.....	31
2.5.2.1 Tamaño de la pantalla.....	32
2.5.2.2 Colores.....	33
2.5.2.3 Texto.....	33
2.5.2.4 Dispositivo de interacción.....	34
2.5.2.5 Navegación.....	34
2.6 CONSIDERACIONES FINALES.....	34

CAPÍTULO 3. DISEÑO DE APLICACIONES INTERACTIVAS

3.1 APLICACIONES INTERACTIVAS EN AMBIENTES GINGA-NCL.....	38
3.1.1 NCL.....	38
3.1.1.1 Nested Context Model.....	38
3.2 APLICACIONES INTERACTIVAS EN AMBIENTES GINGA-J.....	42
3.2.1 CICLO DE VIDA XLETS.....	42
3.3 AMBIENTE GRÁFICO.....	45
3.4 HERRAMIENTAS DE DESARROLLO.....	45
3.4.1 COMPOSER.....	45
3.4.2 GINGAWAY.....	46
3.4.3 LUACOMP.....	47
3.4.4 TVISION.....	48
3.4.5 NCL ECLIPSE.....	49
3.5 DIGITALTV.....	52
3.5.1 DIAGRAMA DE CASOS DE USO.....	54
3.5.2 MENÚ PRINCIPAL.....	55
3.5.2.1 Requisitos.....	55

3.5.2.1.1 Requisitos del Usuario.....	55
3.5.2.1.2 Requisitos del sistema.....	56
3.5.2.2 Diagramas NCM.....	56
3.5.2.3 Diagramas de desarrollo.....	68
3.5.2.4 Implementación.....	70
3.5.3 APLICACIÓN 1. ALIMENTATE ECUADOR.....	70
3.5.3.1 Requisitos.....	70
3.5.3.1.1 Requisitos del Usuario.....	70
3.5.3.1.2 Requisitos del sistema.....	70
3.5.3.2 Diagramas NCM.....	71
3.5.3.3 Diagramas de desarrollo.....	71
3.5.3.4 Implementación.....	72
3.5.4 APLICACIÓN 2. HISTORIA DEL ECUADOR.....	72
3.5.4.1 Requisitos.....	72
3.5.4.1.1 Requisitos del usuario.....	72
3.5.4.1.2 Requisitos del sistema.....	74
3.5.4.2 Diagramas NCM.....	74
3.5.4.3 Diagramas de desarrollo.....	74
3.5.4.4 Implementación.....	75
3.5.5 APLICACIÓN 3. MINISTERIOS.....	76
3.5.5.1 Requisitos.....	76
3.5.5.1.1 Requisitos del usuario.....	76
3.5.5.1.2 Requisitos del sistema.....	77
3.5.5.2 Diagramas NCM.....	78
3.5.5.3 Diagramas de desarrollo.....	78
3.5.5.4 Implementación.....	79
3.5.6 APLICACIÓN 4. GAMATV.....	80
3.5.6.1 Requisitos.....	80
3.5.6.1.1 Requisitos del usuario.....	80
3.5.6.1.2 Requisitos del sistema.....	80
3.5.6.2 Diagramas NCM.....	81
3.5.6.3 Diagramas de desarrollo.....	81
3.5.6.4 Implementación.....	81

3.5.7 APLICACIÓN 5. UPS.....	82
3.5.7.1 Requisitos.....	82
3.5.7.1.1 Requisitos del usuario.....	82
3.5.7.1.2 Requisitos del sistema.....	84
3.5.7.2 Diagramas NCM.....	84
3.5.7.3 Diagramas de desarrollo.....	84
3.5.7.4 Implementación.....	84
3.6 CANAL DE RETORNO.....	85
3.7 CONSIDERACIONES FINALES.....	89

CAPÍTULO 4. SIMULACIÓN Y PRUEBAS

4.1 INTRODUCCIÓN.....	90
4.2 PRUEBAS.....	91
4.2.1 PRUEBAS FUNCIONALES.....	91
4.2.1.1 Casos de prueba.....	91
4.2.1.1.1 Caso de prueba: Menú principal.....	92
4.2.1.1.2 Caso de prueba: Aliméntate Ecuador.....	93
4.2.1.1.3 Caso de prueba: Historia del Ecuador.....	94
4.2.1.1.4 Caso de prueba: Entidades Públicas.....	95
4.2.1.1.5 Caso de prueba: GamaTv.....	96
4.2.1.1.6 Caso de prueba: Ups.....	97
4.3 SIMULACIÓN.....	98
4.3.1 EMULADOR GINGA-NCL.....	98
4.3.2 INSTALACIÓN NATIVA.....	98
4.3.3 MÁQUINAS VIRTUALES.....	100
4.3.3.1 VirtualBox.....	101
4.3.3.1.1 Instalar software de virtualización.....	101
4.3.3.2 VMware.....	102
4.3.3.2.1 Instalar software de virtualización.....	102
4.3.4 PAQUETES DEB.....	104
4.3.5 LIVECD.....	104
4.4 RESULTADOS.....	108
4.5 CONTROL REMOTO.....	114

4.5.1 FUNCIONES NUMÉRICAS.....	114
4.5.2 FUNCIONES INTERACTIVAS.....	114
4.5.3 BOTONES DE COLORES (ROJO, VERDE, AMARILLO Y AZUL).....	115
4.6 ESQUEMA DE TELEVISIÓN DIGITAL INTERACTIVA PARA GAMATV.....	116
4.6.1 BLOQUE DE EMISIÓN.....	117
4.6.1.1 Servidor de aplicaciones(Application Server).....	119
4.6.2 FUNCIONALIDADES DEL SERVIDOR DE APLICACIONES.....	119
4.6.3 TRANSMISIÓN DE DATOS.....	121
4.6.4 BLOQUE DE EDICIÓN Y DESARROLLO.....	121
4.6.5 BLOQUE DE PRUEBAS DE APLICACIONES.....	122
4.7 CONSIDERACIONES FINALES.....	122

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES.....	124
5.2 RECOMENDACIONES.....	126

Anexo A. Entorno de Desarrollo

Anexo B. Lenguaje NCL

Anexo C. Lenguaje Lua

Anexo D. Compilación e Instalación de Ginga-NCL

Anexo E. Compilación e Instalación de OpenGinga

Anexo F. Sistema de Televisión Digital Terrestre ISDB-Tb

Anexo G. Manual del Usuario

Anexo H. EITV Playout Professional

Anexo I. EITV Playout

ÍNDICE DE FIGURAS

Figura 1-1: Esquema televisión digital por satélite.....	6
Figura 1-2: Esquema televisión digital terrestre.....	7
Figura 2-1: Señal Analógica vs. Señal Digital.....	18
Figura 2-2: La televisión digital terrestre interactiva.....	19
Figura 2-3: Visión general del Sistema de Televisión Digital.....	20
Figura 2-4: Arquitectura en capas de la Televisión Digital Terrestre.....	22
Figura 2-5: Arquitectura Ginga.....	24
Figura 2-6: Subsistema Ginga-NCL.....	25
Figura 2-7: Subsistema Ginga-J.....	27
Figura 2-8: Subsistema Ginga-CC.....	29
Figura 3-1: Dimensiones en el desarrollo de aplicaciones interactivas.....	35
Figura 3-2: Diagrama con opciones de desarrollo.....	37
Figura 3-3: Pantalla mostrando varias Xlets.....	38
Figura 3-4: ¿Dónde?.....	39
Figura 3-5: ¿Cómo?.....	40
Figura 3-6: ¿Qué?.....	41
Figura 3-7: ¿Cuándo?.....	41
Figura 3-8: Estados de un Xlet.....	42
Figura 3-9: Modelo para la presentación de las aplicaciones interactivas.....	45
Figura 3-10: Herramienta Composer.....	46
Figura 3-11: Herramienta Gingaway.....	47
Figura 3-12: Herramienta LuaComp.....	48
Figura 3-13: Herramienta Tvision.....	49
Figura 3-14: Herramienta NCL Eclipse.....	50
Figura 3-15: Ciclo de vida de la aplicación.....	53
Figura 3-16: Diagrama de Casos de Uso general de la aplicación.....	55
Figura 3-17: Leyenda de los diagramas NCM.....	57
Figura 3-18: Diagrama NCM del momento inicial de la aplicación.....	58
Figura 3-19: Diagrama NCM, Contexto Menú, opción Aliméntate Ecuador.....	60
Figura 3-20: Diagrama NCM, Contexto Menú, opción Historia del Ecuador.....	60
Figura 3-21: Diagrama NCM, Contexto Menú, opción Entidades Públicas.....	62
Figura 3-22: Diagrama NCM, Contexto Menú, opción Ups.....	63

Figura 3-23: Diagrama NCM, contexto Menu, opción Misión.....	64
Figura 3-24: Diagrama NCM, contexto Menu, opción Visión.....	65
Figura 3-25: Diagrama NCM, contexto Menu, opción Regresar.....	66
Figura 3-26: Diagrama NCM, Contexto Menú, opción GamaTv.....	67
Figura 3-27: Diagrama NCM, Contexto Menú, opción Salir.....	68
Figura 3-28: Localización espacial de la opción del menú de la aplicación.....	70
Figura 3-29: Diagrama de opciones de desarrollo, opción menú.....	70
Figura 3-30: Diagrama de opciones de desarrollo, opción Aliméntate Ecuador....	72
Figura 3-31: Diagrama NCM, contexto Aliméntate Ecuador.....	74
Figura 3-32: Diagrama NCM de la aplicación Historia del Ecuador.....	76
Figura 3-33: Diagrama de opciones de desarrollo, Historia del Ecuador.....	77
Figura 3-34: Diagrama NCM de la aplicación Entidades Públicas.....	79
Figura 3-35: Diagrama de opciones de desarrollo, Entidades Públicas.....	80
Figura 3-36: Diagrama NCM de la aplicación GamaTv.....	84
Figura 3-37: Diagrama de opciones de desarrollo, opción GamaTv.....	84
Figura 3-38: Diagrama de opciones de desarrollo, opción Ups.....	86
Figura 3-39: Diagrama NCM de la aplicación Ups.....	87
Figura 3-40: Posibles formas de canales de comunicación.....	88
Figura 3-41: Sistema de transmisión y recepción del estándar ISDBT-t.....	89
Figura 4-1: Emulador de Ginga-NCL. TeleMidia.....	100
Figura 4-2: Kubuntu con implementación de Ginga de Lifa en VirtualBox.....	103
Figura 4-3: Fedora con implementación de Ginga de Lifa en Vmware.....	104
Figura 4-4: Live Cd implementación de Ginga.....	106
Figura 4-5: Menú principal. Aliméntate Ecuador Seleccionada.....	109
Figura 4-6: Aplicación Aliméntate Ecuador.....	110
Figura 4-7: Aplicación Aliméntate Ecuador, plato seleccionado.....	110
Figura 4-8: Historia del Ecuador, primera pregunta.....	111
Figura 4-9: Historia del Ecuador, resultados, aciertos y errores.....	111
Figura 4-10: Entidades Públicas, Web Service Actas de Finiquito del MRL.....	112
Figura 4-11: Aplicación Entidades Públicas, visualización de la búsqueda.....	112
Figura 4-12: Menú opciones Universidad Politécnica Salesiana.....	113
Figura 4-13: Universidad Politécnica Salesiana, misión.....	113
Figura 4-14: Universidad Politécnica Salesiana, visión.....	114

Figura 4-15: GamaTv, interactividad con el servicio twitter.....	114
Figura 4-16: Control Remoto típico para según la norma brasileña.....	116
Figura 4-17: Esquema de una estación televisiva preparada para la Tv Digital. .	119
Figura 4-18: Componentes de servidor de aplicaciones.....	121
Figura A-1: Instalando Plugin NCL en Eclipse.....	136
Figura A-2: Completando la información sobre el plugin NCL.....	136
Figura A-3: Creando un nuevo documento NCL.....	137
Figura A-4: Configurando un intérprete Lua.....	138
Figura B-1: Estructura básica de un documento NCL.....	141
Figura B-2: Atributos de posición y tamaño de una región.....	144
Figura B-3: Ilustración de un conector causal (elemento <causalConnector>). .	154
Figura B-4: Ilustración de un enlace.....	155
Figura F-1: Bloques funcionales del sistema.....	192
Figura F-2: Ejemplo de configuración de capas.....	194
Figura F-3: Configuración básica del transmisor.....	194
Figura F-4: Selección de codificación de canal.....	195
Figura F-5: Ejemplo de mapeo modulación de 16QAM.....	197
Figura F-6: Decalaje de frecuencia de canal.....	201
Figura F-7: Máscara de transmisión.....	201
Figura G-1: Aviso de interactividad.....	205
Figura G-2: Menú principal de la aplicación.....	206
Figura G-3: Aspecto de las opciones del menú.....	206
Figura G-4: Aliméntate Ecuador, opciones de platos.....	207
Figura G-5: Mensaje al escoger una opción.....	208
Figura G-6: Historia del Ecuador, pregunta con la opción tres seleccionada.....	209
Figura G-7: Historia del Ecuador, mostrando resultados.....	210
Figura G-8: Entidades Públicas. Ingresando un número de documento.....	211
Figura G-9: Información de un turno del Sistema de Actas de Finiquito.....	212
Figura G-10: Segundo menú de la aplicación Ups.....	213
Figura G-11: Opción Misión del segundo menú de la aplicación Ups.....	214
Figura G-12: Opción Visión del segundo menú de la aplicación Ups.....	214
Figura G-13: Aplicación GamaTv, mostrando mensajes de la cuenta de twitter..	215

ÍNDICE DE TABLAS

Tabla 1-1: Estándares de televisión digital terrestre.....	12
Tabla 2-1: Diferencias Televisor y Computador.....	32
Tabla 3-1: Comparativa entre herramientas de desarrollo.....	52
Tabla 4-1: Formato de descripción de casos de prueba.....	92
Tabla 4-2: Descripción de caso de prueba Menú principal.....	92
Tabla 4-3: Descripción de caso de prueba Aliméntate Ecuador.....	93
Tabla 4-4: Descripción de caso de prueba Historia del Ecuador.....	94
Tabla 4-5: Descripción de caso de prueba Entidades Públicas.....	95
Tabla 4-6: Descripción de caso de prueba GamaTv.....	96
Tabla 4-7: Descripción de caso de prueba Ups.....	97
Tabla 4-8: Comparativa Implementaciones Ginga.....	106
Tabla 4-9: Comparativa, funcionalidad y características entre Ginga Live CD y Ginga VMware.....	107
Tabla 4-10: Especificaciones para el control remoto.....	116
Tabla B-1: Parámetros que pueden ser utilizados en descriptor, de acuerdo al archivo multimedia.....	148
Tabla B-2: Tipos de archivo multimedia.....	149
Tabla B-3: URI válidas.....	150
Tabla B-4: Papeles predefinidos de condición.....	154
Tabla B-5: Papeles predefinidos de acción.....	155
Tabla C-1: Palabras reservadas del lenguaje Lua.....	160
Tabla F-1: Tabla del factor de normalización.....	197
Tabla F-2: Resumen de las características técnicas.....	198
Tabla G-1: Tabla de opciones de interacción del menú.....	205
Tabla G-2: Tabla de opciones para la aplicación Aliméntate Ecuador.....	206
Tabla G-3: Tabla de opciones para la aplicación Aliméntate Ecuador.....	208
Tabla G-4: Tabla de opciones para la aplicación Entidades Públicas.....	210
Tabla G-5: Tabla de opciones para la aplicación Ups.....	212
Tabla G-6: Tabla de opciones para la aplicación GamaTv.....	214

RESUMEN

La televisión es el medio de comunicación más usado a nivel mundial, la televisión permite acceder a información, entretenimiento, cultura, etc. Muchos beneficios han llegado con la televisión digital, alta definición, calidad en la señal de video y audio, interactividad, etc. La interactividad permite ofrecer servicios que no eran posibles con la televisión tradicional, sobre todo en la variedad de servicios computacionales a través de la TV. En el año 2010 el gobierno ecuatoriano eligió el estándar de televisión digital Sbtvd para nuestro país, estándar japonés con variantes desarrolladas por investigadores brasileños, adaptando la tecnología a la realidad de su país, realidad que no dista de nuestra sociedad, ni tampoco de toda la sociedad latinoamericana. Dentro de las innovaciones realizadas al estándar brasileño se encuentra el *Middleware* para desarrollar aplicaciones interactivas, Ginga es el nombre de este componente, que permite ejecutar aplicaciones interactivas en ambientes de presentación Ginga-NCL y en ambientes de ejecución Ginga-J. Este trabajo de investigación es un estudio sobre *Middleware* del estándar Sbtvd, una guía de desarrollo de aplicaciones interactivas para televisión digital y el diseño de una aplicación para GamaTv donde se intenta demostrar todo el potencial de la interactividad de la era de la televisión digital.

PRESENTACIÓN

Con la llegada del nuevo estándar de televisión digital terrestre a nuestro país, la era de la televisión analógica está destinada a ser reemplazada. Esta transición afronta todos los retos que conlleva pasar de una tecnología a otra totalmente nueva y diferente, y que en el caso de la televisión digital terrestre, son innumerables los desafíos que trae consigo para todas las personas y empresas que están envueltas de una u otra forma en el mundo de las telecomunicaciones audio visuales.

Las autoridades estatales tienen el trabajo de establecer el marco jurídico, regulatorio, técnico para lograr esta transición de manera exitosa. Las personas tendrán que comprar televisores que funcionen con el nuevo estándar o en su defecto adquirir decodificadores, las productoras deberán crear contenido bajo los nuevos parámetros de la televisión digital, las estaciones de televisión deberán invertir en equipos y adaptar su visión hacia las nuevas tecnologías.

La televisión digital terrestre trae consigo la posibilidad de ofrecer a los usuarios servicios novedosos como concursos o juegos e importantes servicios como inclusión social o gobierno electrónico con las llamadas aplicaciones interactivas, esto abre un gran espectro de oportunidades para el desarrollo de nuevos y variados negocios con la participación de actores anteriormente excluidos.

Pero al ser un estándar totalmente nuevo y en proceso de implementación, el reto más grande a los cuales se enfrentará GamaTv es adaptarse a las nuevas características interactivas tales como:

- ¿Cuál es el procedimiento para desarrollar aplicaciones interactivas?
- ¿Qué tipo de aplicación interactiva desarrollar?
- ¿Qué clase de herramientas son necesarios para desarrollar aplicaciones interactivas?
- ¿Cómo ofrecer servicio y entretenimiento en una aplicación interactiva?

- ¿Cómo adaptar una aplicación interactiva al contenido de la programación audio visual?
- ¿Qué licenciamiento adquiere una aplicación interactiva con el estándar SBTVD?

Sin un lineamiento claro es muy obvio que GamaTv no estará preparada para ofrecer a sus fieles televidentes servicios interactivos innovadores y de gran calidad, por lo que esta empresa se quedaría al margen de estas innovaciones tecnológicas, lo que conllevaría a la pérdida de niveles de sintonía.

Si bien la televisión es un servicio, su manejo y administración debe entenderse como un negocio, por lo que en ausencia de estas innovaciones, las pérdidas económicas por este concepto serían demasiado grandes, sin dejar de lado la inversión que se necesitará hacer en equipamiento y su sobre dimensionamiento al no aprovechar todas las características que el estándar SBTVD proporciona, las oportunidades de crecimiento y fortalecimiento de la empresa se verían comprometidas y estancadas.

Es de suma importancia integrar en estas aplicaciones interactivas el tipo de información necesaria para uso del televidente y la facilidad para lograr sus propósitos, esto debe ir de la mano con la calidad de los desarrollos, temas específicos que no existen o no están claros en GamaTv y que esta investigación abordará y esclarecerá.

OBJETIVOS

Objetivo general

Diseñar Aplicaciones Interactivas T-Government, T-Health y T-Learning para su aplicación en el Sistema de Televisión Digital Terrestre del Ecuador(SBTVD) para la empresa Televisión del Pacífico S.A. GamaTV.

Objetivos específicos

- Investigar el nuevo estándar seleccionado por el país para la televisión digital terrestre SBTVD.
- Examinar el surgimiento del middleware de desarrollo Ginga, sus diferentes implementaciones Ginga NCL, Ginga-J, y cuál de ellas es la mejor opción para adaptarla a un desarrollo específico.
- Documentar como el software libre garantiza el acceso a la información de todo el proceso evolutivo haciendo posible su adaptación a la realidad de cada país.
- Desarrollar Aplicaciones Interactivas T-Goverment, T-Learning y T-Healt.
- Probar el correcto funcionamiento de las Aplicaciones Interactivas T-Goverment, T-Learning y T-Healt en un ambiente de simulación.
- Establecer un estándar para el desarrollo de Aplicaciones Interactivas de la Televisión Digital Terrestre para GamaTV.
- Compartir las experiencias y resultados del diseño de Aplicaciones Interactivas con la comunidad Ginga como parte de la filosofía del software libre.

CAPÍTULO 1. MARCO TEÓRICO

1.1 SÍNTESIS HISTÓRICA DE LA TELEVISIÓN

1.1.1 HISTORIA DE LA TELEVISIÓN EN EL MUNDO

En el año 1884, Paul Gottlieb Nipkow inventó un dispositivo mecánico para analizar imágenes de manera ordenada, lo que conocemos como exploración de imágenes en la televisión actual.

Alrededor del año 1904 Alan Archibald Campbell-Swinton y Boris Lvovich Rosing trabajaban paralelamente en sus investigaciones, ambos inventores utilizaron sistemas mecánicos para la captura de imágenes, y como receptor el uso de tubo de rayos, este fue el primer precedente del empleo de este dispositivo para propósitos de la invención de un sistema de televisión.

Como resultado de estas investigaciones nacen dos sistemas totalmente diferentes: la televisión mecánica y la televisión electrónica.

1.1.1.1 La televisión mecánica

En 1923 el inventor Charles Jenkins usando el disco de Nipkow inventa el primer sistema práctico de televisión mecánica. En 1926, John Logie Baird perfeccionó el disco de Nipkow y logró transmitir imágenes con movimiento a través del sistema de disco mecánico, es así como la ciudad de Londres vivió este acontecimiento protagonizado por la empresa Británica British Broadcasting Corporation BBC.

La televisión mecánica tuvo grandes avances y sus investigadores seguían agregando muchas innovaciones, pero en el año 1934 los sistemas de televisión se habían convertido al sistema electrónico, por su calidad de imagen es el que se utiliza hasta hoy en día.

1.1.1.2 La televisión electrónica

Las investigaciones de Alan Archibald Campbell-Swinton con los tubos de rayos catódicos contribuyeron a los primeros pasos de la televisión electrónica pero no

fue suficiente para convertirlo en realidad. En el año de 1927 el inventor Philo Taylor Farnsworth fue capaz de inventar un modelo funcional de televisión electrónica basado en las ideas de Switon.

Desde muy pequeño Farnsworth sorprendía a sus maestros de escuela con la idea de trasladar las imágenes a un aparato electrónico, a sus 21 años de edad haría su más grande sueño, inventar un sistema de televisión, los sistemas de discos mecánicos quedaron al margen y su invento es la base de lo que tenemos en la actualidad.

Nacen las principales empresas de la industria, entre las cuales se destacan la BBC en Gran Bretaña y la NBC en Estados Unidos, países como Francia, Alemania, Italia, la ex Unión Soviética comienzan a realizar transmisiones en sus principales ciudades, la explosión de la segunda guerra mundial detuvo su desarrollo y fue luego de ésta que los gobiernos de los principales países retomaron sus investigaciones.

La televisión a color presentó un nuevo reto para los científicos e inventores, fue en la década de los años 50 que en Estados Unidos se presentó el estándar NTSC, compatible con los receptores monocromáticos, Francia creó su estándar llamado SECAM y Alemania hizo lo mismo con su estándar PAL, cada país hacía su elección y de este modo la televisión se expandía de manera acelerada por el mundo.

En América, México y Brasil empiezan sus transmisiones en el año 1951, Argentina un año más tarde, en 1952 es el turno de Venezuela y Perú en el año 1958.

Desde su nacimiento la televisión ha servido de puente comunicativo para dar a conocer a la gente hechos de gran importancia.

1.1.2 HISTORIA DE LA TELEVISIÓN EN ECUADOR

El 11 de julio de 1959 arriban a Quito equipos de transmisión de televisión

provenientes de las bodegas de General Electric en Syracuse, New York, un norteamericano de apellido Hartwell luego de repararlos, los dona a la emisora de radio HCJB, para luego ser exhibidos en la feria estudiantil del colegio Americano, los asistentes pudieron apreciar la televisión en blanco y negro por primera vez en nuestro país.

Con la llegada de los primeros equipos de televisión profesional a la ciudad de Guayaquil, José Rosenbaum y su esposa Linda Zambrano llegan a un acuerdo con la Casa de la Cultura de esta ciudad y se funda la “Compañía de Televisión Ecuatoriana”, el 12 de diciembre de 1960 se transmite la primera señal comercial del Ecuador a través del canal 4, en lo que hoy conocemos como Red Telesistema.

En el año de 1967 nace Ecuavisa desde el Cerro del Carmen en Guayaquil y en Cuenca el canal 3 hace su aparición desde el centro mismo de la ciudad. Para el año 1969 Telecentro emitió su programación desde el Puerto Principal en el canal 10.

Más tarde con la llegada de la televisión a color en la década de los 70, nace Televisora del Amazonas, Televisión del Pacífico (GAMATV) inicia sus emisiones en el año de 1976 con su matriz ubicada en la ciudad de Quito y con cobertura a nivel nacional, a partir de 1980 las demás televisoras se expanden por las principales ciudades del Ecuador.

En 18 de octubre del año 2007 mediante Decreto Ejecutivo N.681, se delega a la Superintendencia de Telecomunicaciones el análisis, las pruebas y recomendaciones para la inclusión de nuevas tecnologías de radiodifusión y televisión.

El 21 de abril de 2008 se firma un acuerdo de cooperación técnica e instrumental entre la Superintendencia de Telecomunicaciones y la embajada de Japón en Ecuador, con el afán de coordinar el préstamo de los equipos necesarios para realizar las respectivas pruebas del estándar ISDB-T. De esta manera la

Superintendencia de Telecomunicaciones conforma un equipo técnico para la ejecución de pruebas y evaluación de los distintos estándares internacionales.

Los días 4 y 5 de diciembre de 2008 en la ciudad de Cuenca se realizó el primer foro internacional de “Televisión Digital Terrestre en América Latina y El Caribe” con la participación de países de la región que todavía no habían adoptado un estándar y la presencia de los representantes de las tecnologías ATSC, DVB-T, ISDB-T y DTMB que pusieron a consideración de los asistentes las características de cada uno de sus sistemas. En este evento se realizó una pequeña demostración del estándar Japonés-Brasileño con equipos móviles de transmisión y recepción.

El 09 de diciembre de 2008 se realiza la primera transmisión de televisión digital terrestre en la ciudad de Quito, el equipo transmisor del estándar ISDB-T fue instalado en el cerro Pichincha y se realizó las pruebas de recepción desde las oficinas centrales de la Superintendencia de Telecomunicaciones. Posteriormente llegó un equipo de transmisión del estándar DVB-T en calidad de préstamo por parte de la Unión Europea, para el 20 de febrero del 2009 se inician oficialmente en la ciudad de Quito la etapa de pruebas del estándar Japonés-Brasileño y el estándar Europeo en distintos puntos de la ciudad capital. A finales del mes de junio del mismo año, se llevan acabo las pruebas de rendimiento del estándar chino DTMB.

Finalmente la Superintendencia de Telecomunicaciones entrega el informe de definición del estándar de televisión digital terrestre anunciando su recomendación hacia el sistema Japonés-Brasileño ISDB-T el mismo que es aprobado por el CONATEL¹ el 25 de marzo de 2010.

1 Consejo Nacional de Telecomunicaciones del Ecuador

1.2 LA TELEVISIÓN DIGITAL

La televisión digital codifica sus señales de forma binaria, haciendo posible la integración de muchos servicios adicionales que no eran posibles con la televisión analógica.

La televisión digital dependiendo del medio y el método de transmisión es posible encontrar de las siguientes formas:

- Televisión digital por satélite
- Televisión digital por cable
- Televisión digital terrestre
- IPTV

1.2.1 TELEVISIÓN DIGITAL POR SATÉLITE

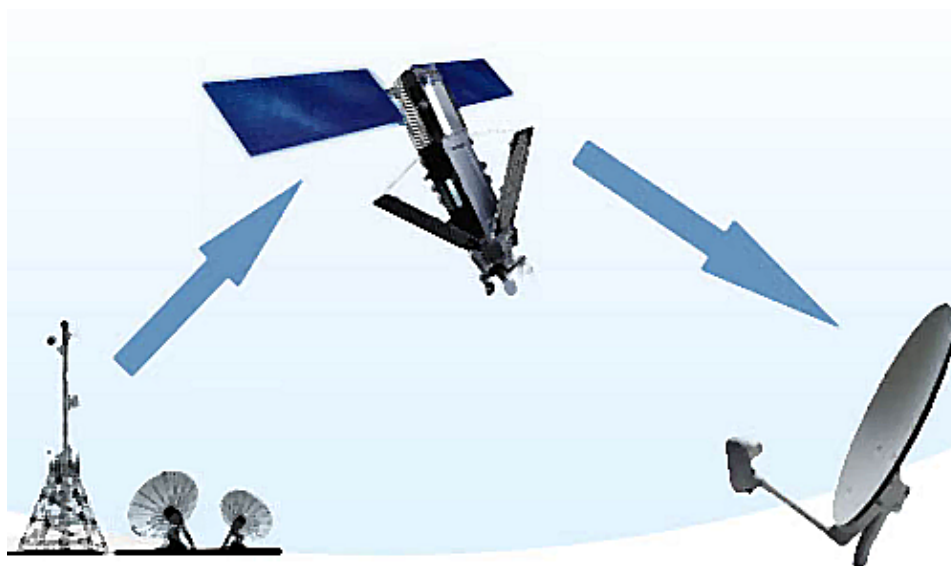
La televisión digital por satélite funciona principalmente con la señal generadora de la información ubicada en cualquier zona de la tierra hacia el receptor en el espacio, también llamada transmisión ascendente, en cambio, la transmisión descendente ocurre cuando la información va desde el satélite en órbita hasta una antena receptora en cualquier lugar del planeta. Este sistema es muy usado para llevar imágenes al momento en que ocurren, en zonas alejadas de la recepción, esto posibilitó que se pueda presenciar eventos de interés casi al momento de que se lleven a cabo. Figura 1-1.

1.2.2 TELEVISIÓN DIGITAL POR CABLE

La televisión digital por cable es el resultado de la integración de muchas tecnologías para facilitar la recepción de la señal de televisión en aquellos lugares que por su ubicación o topografía resulta muy difícil la transmisión por vía aérea. Los principales medios físicos de transmisión son la fibra óptica y el cable coaxial.

Las empresas dedicadas a brindar este servicio instalan sus redes desde su centro de transmisión hasta los hogares de los abonados, por esta razón la televisión digital por cable suele implantarse en lugares con densa población para hacer rentable su negocio.

Figura 1-1: Esquema televisión digital por satélite

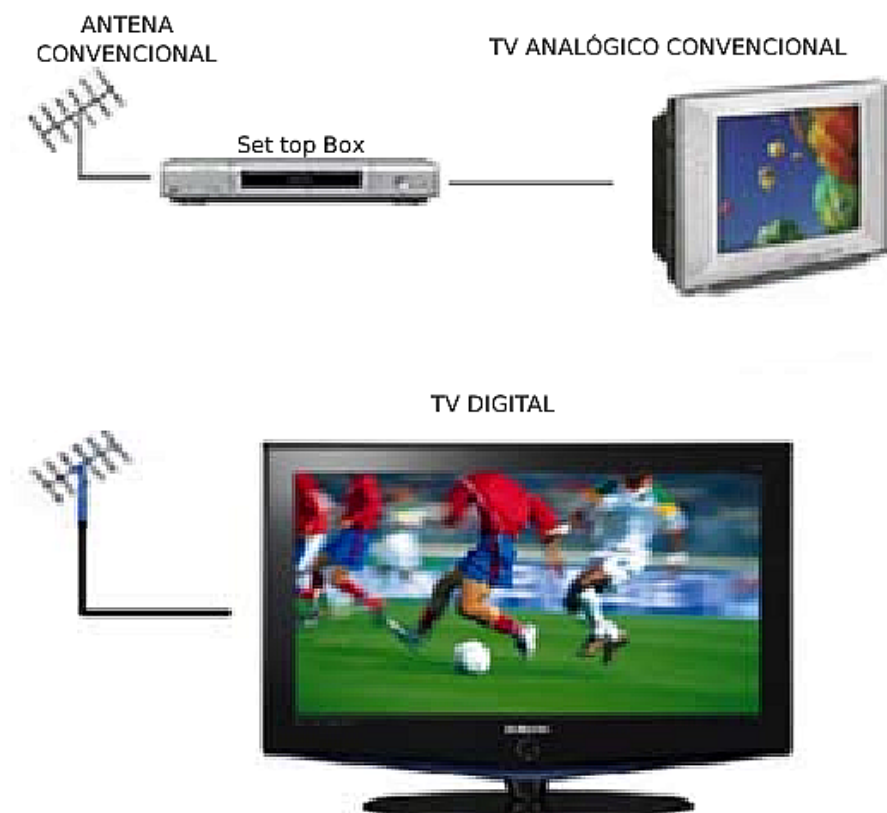


Fuente: <http://www.supersatelliteselection.com/how-satellite-tv-works.html>

1.2.3 TELEVISIÓN DIGITAL TERRESTRE

La televisión digital terrestre es la evolución de la televisión tradicional gratuita a la que se ha podido acceder en el país desde hace mucho tiempo, con la diferencia de que la manera de transmitir la información análogamente será remplazada por métodos digitales, lo que trae consigo grandes novedades en todos los ámbitos, por ejemplo, la calidad de la señal de audio y video mejorará sustancialmente, se añadirán características novedosas y se podrá dar un uso muy diferente a los televisores aprovechando las aplicaciones interactivas que van a ser parte de la vida cotidiana. La televisión digital terrestre es un paso necesario en la convergencia hacia los medios digitales que el mundo está viviendo en todos los campos. Figura 1-2.

Figura 1-2: Esquema televisión digital terrestre



Fuente: SUPERTEL Revista Institucional N. 3, Tv Digital para Ecuador, 2008

1.2.4 IPTV

La IPTV hace uso de la arquitectura de redes, mediante los protocolos de Internet para entregar servicios de televisión a sus clientes, en lugar de los métodos tradicionales de difusión mediante radio frecuencia, satélite o cable. Los abonados pueden solicitar el servicio el momento en que lo deseen, permitiendo personalizar lo que cada cliente quiera ver. Se ofrecen servicios de pago por evento o televisión en vivo o en diferido, lo que también se conoce como video bajo demanda. Se necesitan de infraestructuras de redes de alta velocidad para garantizar la calidad del servicio.

1.3 ESTÁNDARES INTERNACIONALES DE TELEVISIÓN DIGITAL TERRESTRE

Actualmente existen cuatro estándares internacionales para la televisión digital terrestre, el estándar ATSC, ISDB-T, DVB-T y el DTMB. Además de contar con las modificaciones brasileñas hechas al estándar ISDB-T el mismo que es adoptado por el Ecuador y la gran mayoría de países sudamericanos.

1.3.1 ATSC (ADVANCED TELEVISION STANDARDS COMMITTEE)

ATSC son el conjunto de normas creadas para las transmisiones digitales terrestres en los Estados Unidos desarrollado en el año de 1982 por un grupo de empresas privadas dedicadas a la radiodifusión y la fabricación de equipos electrónicos, destinada a reemplazar el sistema análogo NTSC². Entre los principales países que adoptaron este estándar están Canadá, el Salvador, Honduras y México, por citar algunos.

Dentro de las principales características se cita las siguientes:

- Para la compresión de datos usa el sistema de especificación MPEG-2³.
- AC-3⁴ es usado como el códec⁵ de audio lo que permite cinco canales de audio con un sexto canal LFE⁶.
- Emplea la modulación 8VSB⁷, detención y corrección de errores Reed Solomon⁸.

2 National Television System Committee es el sistema de codificación y transmisión de Televisión a color analógico desarrollado en Estados Unidos.

3 Moving Pictures Experts Group 2, es la designación para un grupo de estándares de codificación de audio y vídeo acordado por MPEG (grupo de expertos en imágenes en movimiento), y publicados como estándar ISO 13818

4 Audio Codec 3, es el nombre del códec de audio desarrollada por los laboratorios Doubly.

5 Es la abreviatura de codificador-decodificador.

6 Low Frequency Effects, es comúnmente usado en la descripción de una pista de audio contenida dentro de los efectos de sonido de una película.

7 8-level Vestigial Sideband estándar de modulación de radio frecuencia

8 Estándar de detención y corrección de errores creado por Irving Reed y Gustave Solomon, de ahí su nombre.

- Incluye televisión digital de alta definición(HDTV), multicanal en definición estándar(SDTV) dependiendo del ancho de banda del canal y del sistema de compresión empleado.

1.3.2 DVB-T (DIGITAL VIDEO BROADCASTING - TERRESTRIAL)

El DVB-T es el sistema para la transmisión digital terrestre como parte de la organización DVB, que se encuentra conformado por entidades gubernamentales y empresas privadas principalmente de países de la Unión Europea que empezó su desarrollo en el año 1993. En el año de 1995 se presentó el resultado de las investigaciones dando origen a las normas para las tecnologías de televisión digital satelital(DVB-S), televisión digital por cable(DVB-C), televisión digital mediante ADSL(DVB-IPTV), las emisiones para dispositivos móviles(DVB-H) y la televisión digital terrestre(DVB-T). En la mayoría de países del viejo continente ya se encuentra en uso y paulatinamente irá reemplazando al sistema análogo PAL⁹ y SECAM¹⁰. Este estándar también ha sido adoptado en Uruguay, Colombia y Australia por mencionar algunos países.

Dentro de las principales características se cita las siguientes:

- Para la compresión de datos usa el sistema de especificación MPEG-2 y recientemente se ha incorporado el sistema H.264/MPEG-4¹¹.
- AC-3 es usado como el códec de audio.
- Emplea la modulación COFDM¹², detención y corrección de errores Reed Solomon.
- Incluye televisión digital de alta definición(HDTV), multicanal en definición

9 Phase Alternating Line es el sistema de codificación utilizado en la transmisión de señales de televisión analógica en países africanos, asiáticos europeos y algunos sudamericanos.

10 Es un sistema para la codificación de televisión en color analógica utilizado por primera vez en Francia.

11 H.264/MPEG-4/AVC (Advanced Video Coding), es un estándar de compresión de video desarrollado por la ITU-T Video Coding Experts Group (VCEG) junto con la ISO/IEC Moving Picture Experts Group (MPEG)

12 Coded Orthogonal Frequency Division Multiplexing

estándar(SDTV) dependiendo del ancho de banda del canal y del sistema de compresión empleado.

1.3.3 DTMB (DIGITAL TERRESTRIAL MULTIMEDIA BROADCAST)

Es el estándar de televisión digital terrestre desarrollado por la República Popular China, en el año de 1995 el gobierno encarga las investigaciones del nuevo estándar a los institutos de investigación y las universidades, como resultado de este trabajo se crean dos prototipos¹³ que finalmente se fusionan para crear el estándar de televisión terrestre digital chino en el año 2006. Además de China, Hong Kong y Macao optaron por este sistema.

Dentro de las principales características se cita las siguientes:

- Para la compresión de datos usa el sistema de especificación MPEG-2 y el sistema de especificación H.264/MPEG-4.
- Para la compresión de audio se usa MPEG-1 Layer 2 y también AC3.
- Emplea la modulación TDS-OFDM¹⁴, detención y corrección de errores Reed Solomon.
- La técnica de modulación TDS-OFDM se emplea también para dispositivos móviles, con una cobertura de 10 Km mayor al estándar Europeo y proporcionando una señal aceptable con receptores portátiles moviéndose a velocidades cercanas a los 200 Km/h¹⁵.
- Las transmisiones en alta definición HDTV o varios canales en SDTV, dependiendo del ancho de banda del canal y del sistema de compresión empleado.

13 La Jiaotong University desarrolló el modelo ADTB-T, Tsinghua University desarrolló el modelo DMB-T, juntos dieron lugar al estándar oficial de China.

14 Time Domain Synchronous-Orthogonal frequency-division multiplexing

15 Un desarrollador del estándar DMB-T de la Tsinghua University asegura estas características del estándar chino en base a sus propias pruebas de desempeño.

1.3.4 ISDB-T (INTEGRATED SERVICES DIGITAL BROADCASTING - TERRESTRIAL)

El ISDB-T es el estándar de televisión terrestre digital, parte del conjunto de normas creadas por Japón para definir todas las transmisiones digitales tanto para radio como para televisión, los estándares provenientes del ISDB se dividen dependiendo de la tecnología que usen, así para la televisión digital terrestre ISDB-T, para la televisión digital satelital ISDB-S, para la televisión digital por cable ISDB-C, multimedia ISDB-Tmm y la radio digital ISDB-Tsb. El estado brasileño usó este sistema como base para implementar algunos cambios que dieron lugar a la implementación SBTVD¹⁶ o ISDB-Tb¹⁷.

Dentro de las principales características se cita las siguientes:

- Para la compresión de datos usa el sistema de especificación H.264/MPEG-2.
- MPEG-2 AAC¹⁸ es usado como códec de audio.
- Emplea la modulación COFDM, detención y corrección de errores Reed Solomon.
- Las transmisiones se las puede realizar en alta definición(HDTV) o multicanal en definición estándar, dependiendo del ancho de banda del canal y del sistema de compresión empleado.

1.3.5 ISDB-Tb

Los institutos de investigación principalmente las universidades brasileñas conjuntamente con científicos del país sudamericano introdujeron, algunas características al modelo original, el grupo japonés DiBEG responsable de la norma las adoptó re nombrando al estándar en ISDB-T International.

16 Sistema Brasileiro de Televisão Digital, usado para denominar al estándar brasileño en lengua portuguesa.

17 Integrated Services Digital Broadcasting - Terrestrial Brazilian version

18 Advanced Audio Coding es un estándar de compresión para audio digital, parte de las especificaciones MPEG-2 y MPEG-4.

Dentro de los cambios incorporados al estándar japonés se cita las siguientes:

- Para la compresión de datos usa el sistema de especificación H.264/MPEG-4 en lugar del MPEG-2.
- MPEG-4 AAC v2¹⁹ es usado como códec de audio.
- Emplea la modulación BST-OFDM²⁰.
- Middleware²¹ de desarrollo de aplicaciones interactivas Ginga

Los países sudamericanos que se sumaron a la adopción de este estándar son Argentina, Chile, Perú, Venezuela y Ecuador.

A continuación en la Tabla 1-1 se puede apreciar de mejor manera cada una de las características más importantes de cada estándar de televisión digital terrestre.

Tabla 1-1: Estándares de televisión digital terrestre

	Modulación	Compresión		Interactividad	
		Video	Audio	Declarativo	Procedural
ATSC	8VSB	MPEG-2	AC-3	ACAP-X	ACAP-J
DVB-T	COFDM	MPEG-2	AC-3	DVB-HTML	MHP
DTMB	TDS-OFDM	MPEG-2	AC-3		
ISDB-T	COFDM	MPEG-2	AAC	BML	ARIB-AE
ISDB-Tb	BST-OFDM	MPEG-4	AACv2	GINGA-NCL	GINGA-J

Fuente: El autor

¹⁹ Advanced Audio Coding versión 2.

²⁰ Band Segmented Transmission-Orthogonal Frequency Division Multiplexing

²¹ En términos de televisión digital se puede definir al middleware como el ambiente que abstrae la arquitectura de hardware y la del sistema operacional de los receptores.

1.3.5.1 Adopción del estándar ISDB-Tb en Ecuador

La Superintendencia de Telecomunicaciones presentó al Conatel²² el informe²³ de la definición del estándar de televisión digital terrestre para el Ecuador, en el que se estableció detalladamente los procedimientos llevados a cabo para el análisis de cada uno de los sistemas internacionales y los resultados que determinaron la elección ISDB-Tb como la mejor opción para el país.

El informe establece la siguiente estructura de organización para el proceso de adopción e implementación del sistema de televisión digital terrestre:

- Estudio y pruebas técnicas.
- Estudio de usos, hábitos y preferencias de la televisión en el Ecuador.
- Impacto socio-económico de la definición e implementación de la televisión digital terrestre en el Ecuador.
- Cooperación y transferencia tecnológica.
- Políticas e integración.
- Análisis regulatorio.
- Comunicación y socialización del proceso.
- Evaluación.
- Conclusiones y recomendaciones.

Luego de un poco más de un año, al analizar cada estándar internacional a excepción del ATSC²⁴, el informe es presentado al Conatel, el mismo que acepta

²² Consejo Nacional de Telecomunicaciones del Ecuador.

²³ El informe de definición del estándar de televisión digital terrestre se encuentra disponible en la página web de la Superintendencia de Telecomunicaciones.

²⁴ Según la Superintendencia de Telecomunicaciones, los responsables del estándar ATSC no entregaron a tiempo los equipos necesarios para realizar las pruebas.

la sugerencia de la Superintendencia de Telecomunicaciones y se establece el sistema ISDB-Tb para el país.

El informe en su parte final después de todos los estudios a los que fueron sometidos pone a consideración el orden de prelación de los estándares de televisión digital terrestre:

1. ISDB-T/SBTVD

2. DVB-T

3. DTMB

4. ATSC

El Conatel firma los acuerdos de cooperación con los representantes del estándar ISDB-Tb entre las principales autoridades de Ecuador, Brasil y Japón.

1.4 APAGÓN DE LA TELEVISIÓN ANALÓGICA EN ECUADOR

Después de que la decisión sobre el estándar de televisión digital terrestre ha sido tomada, se establecen los pasos a seguir, lo que viene en el futuro es un camino largo y complicado, ya que, la implementación de una tecnología tan compleja traerá consigo muchas inquietudes y preguntas por resolver, por tanto, las autoridades deben organizar de la mejor manera este proceso para que llegue a su feliz término. Se pueden diferenciar claramente tres escenarios distintos a través de los cuales con las decisiones correctas harán exitoso la migración hacia la televisión digital.

- La ley de radiodifusión y televisión
- Las empresas televisivas
- El consumidor final

1.4.1 LA LEY DE RADIODIFUSIÓN Y TELEVISIÓN

En el marco regulatorio es muy importante que se dé trámite a la nueva normativa que de paso ha la implementación del estándar de televisión digital terrestre. La reorganización de todo el espectro radioeléctrico es una de las bases fundamentales para emprender el camino correcto, además estas normas deben contemplar el proceso más adecuado de adjudicación de frecuencia de cada empresa televisiva, establecer las normas técnicas de operatividad y el espacio que los medios comunitarios como nuevos actores del proceso deben cumplir, al aprovechar las características de la tecnología digital esto es perfectamente posible al hacer un uso mucho más adecuado del espectro radioeléctrico, más señales de televisión estarán disponibles para el consumidor final, sin duda las decisiones que se tomen en el aspecto jurídico serán esenciales para asegurar el equitativo acceso a las nuevas tecnologías para todos.

1.4.2 LAS EMPRESAS TELEVISIVAS

La inversión que deben hacer las empresas televisivas privadas está alrededor de los \$1.500 millones de dólares²⁵ en la compra de todos los equipos necesarios para llevar su señal digital hasta los hogares de las familias ecuatorianas. Sin duda una inversión bastante fuerte, que en un plazo de seis a siete años²⁶ se tiene que llevar a cabo, para asegurar la presencia de estas compañías en el sector de las comunicaciones.

La planificación de las autoridades con los representantes de estos medios garantizarán la consecución de las metas.

1.4.3 EL CONSUMIDOR FINAL

Sin duda una parte fundamental en el proceso de migración del sistema de televisión son todos los ecuatorianos y las personas que residen en nuestro país, la televisión es y será parte importante en la vida cotidiana de la gente y para

²⁵ Valor estimado de la Superintendencia de Telecomunicaciones que divulga en su informe.

²⁶ Se estima este plazo en el informe de la Superintendencia de Telecomunicaciones, pero todavía no existe una fecha exacta.

lograr con éxito la adopción de las nuevas tecnologías en el campo de la televisión, es necesario estar informados de cómo, cuándo y del porqué este cambio, sobre todo cuales son los pasos que las autoridades van a dar para garantizar el acceso de todos a la televisión digital terrestre, ya que, la compra de nuevos equipos en este caso un televisor²⁷ nuevo o un Set Top Box²⁸ es muy difícil para las escalas más pobres del Ecuador, el Estado debe asegurar mediante planes gubernamentales, por ejemplo, el acceso al equipamiento básico para poder receptar la futura televisión digital.

27 El televisor debe contar con la tecnología necesaria para poder recibir señales del estándar ISDB-Tb.

28 El Set Top Box es un decodificador de señales de televisión digital que se acopla a un televisor análogo.

CAPÍTULO 2. TELEVISIÓN DIGITAL INTERACTIVA

2.1 INTRODUCCIÓN

Las aplicaciones interactivas para televisión digital no son nuevas, las empresas de televisión de suscripción sea por cable o satelital proporcionan a sus usuarios esta funcionalidad desde hace algún tiempo atrás, usando las mismas tecnologías que ahora se emplean mundialmente en la televisión digital terrestre.

Sin embargo, la realidad actual del país con respecto al acceso a las nuevas tecnologías de telecomunicaciones y de Internet muestra un gran desfase, según la Superintendencia de Telecomunicaciones solo el 14,21% tiene acceso a la Internet²⁹, en contra partida el acceso a la televisión abierta o por suscripción tiene una penetración del 31%³⁰, confirmando que la televisión es un recurso muy favorable para el uso de aplicaciones interactivas con sentido social o educativo.

La interactividad es un término muy utilizado en los ámbitos informáticos, de diseño y multimedia, de donde se explica que la interactividad de un programa es la acción que conlleva a interactuar al usuario con su ordenador.

2.2 TELEVISIÓN DIGITAL

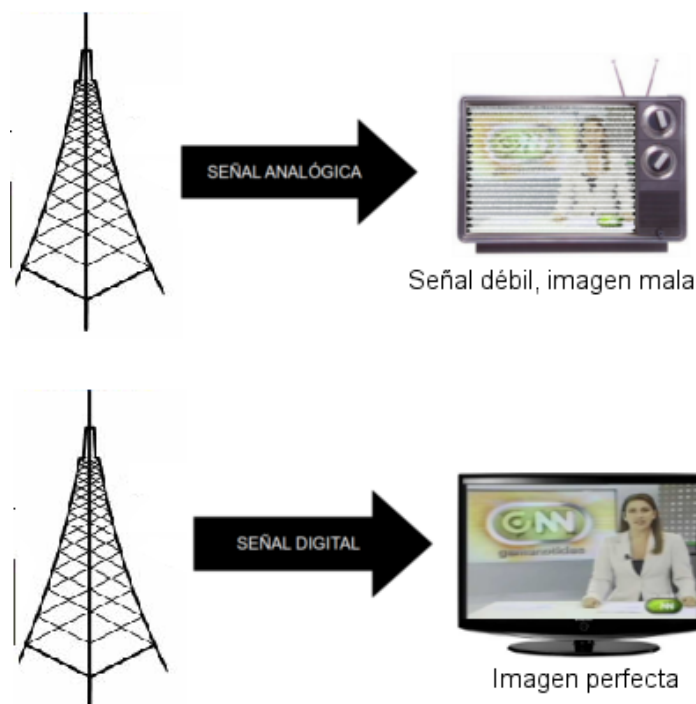
La televisión digital representa una mejor calidad y más servicios con respecto a la televisión analógica, calidad de video y audio, idiomas simultáneos, subtítulos, iguala la cobertura de la señal analógica con menos potencia, mayor número de canales en el mismo ancho de banda, personalización del contenido y principalmente servicios interactivos. La televisión analógica tiene muchos problemas con la calidad de la señal, las interferencias hacen que se produzcan fantasmas³¹ en la imagen (figura 2-1), debido a que no cuentan con un sistema de corrección de errores, este sin duda es uno de los problemas más comunes con esta tecnología.

29 SUPATEL, *Resumen estadístico del No de estaciones de audio y video por suscripción*, Quito, Diciembre 2010

30 <http://www.ecuadorencifras.com/cifras-inec/cienciaTecnologia.html>. Censo 2010 realizado en el país a nivel nacional.

31 También conocido como doble imagen, producida debido a la existencia de un eco en la propagación de la señal analógica de televisión.

Figura 2-1: Señal Analógica vs. Señal Digital



Fuente: SOUZA JÚNIOR, Paulo José, LuaComp: Ferramenta de Autoria de Aplicações para TV Digital, 2009

2.2.1 ESTRUCTURA GENERAL DE LA TELEVISIÓN DIGITAL

A diferencia de la televisión convencional, en la televisión digital la señal que se transmite hasta el usuario final o televidente tiene consigo tres tipos de subflujos: audio, video y datos. Estos subflujos son multiplexados³² para posteriormente ser transmitidos a los receptores (Figura 2-2). Los subflujos de datos también pueden llevar paquetes de control para vigilar las limitantes que tienen las aplicaciones interactivas con respecto al procesamiento y la interfaz del usuario.

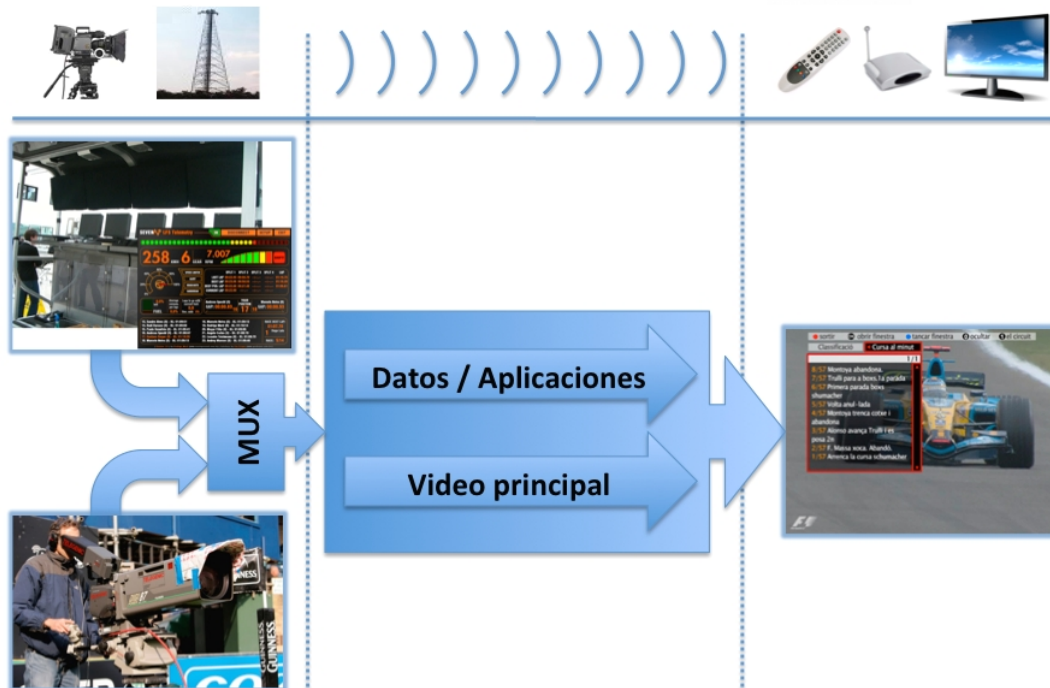
Estas restricciones exigen a los desarrolladores tener cierta precaución al diseñar la aplicación, sus colores, las imágenes, el tamaño de las fuentes, el correcto posicionamiento de los objetos de media³³ y en general la sobrecarga de los

³² La multiplexación es la combinación de dos o más canales de información en un solo medio de transmisión

³³ La Asociación Brasileña de Normas Técnicas establece que los objetos de media son todos los archivos multimedia que puede manejar el middleware Ginga, que pueden ser: video, audio, imágenes, texto, aplicaciones de ejecución.

recursos de hardware.

Figura 2-2: La televisión digital terrestre interactiva



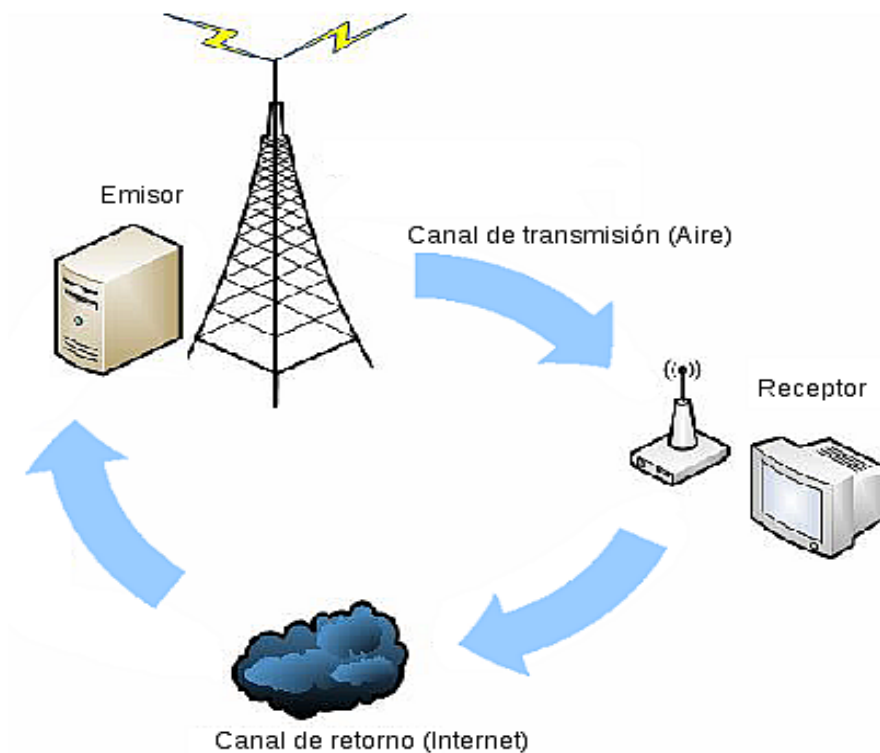
Fuente: FERNÁNDEZ , Alejandro, Televisión Digital Terrestre Interactiva, 2010

De manera general un sistema de Televisión Digital Interactivo puede estar compuesto por:

- **Emisor:** Se encarga de proveer el contenido a los televidentes, así como también el soporte para las aplicaciones interactivas.
- **Receptor:** Presenta el contenido al televidente para que pueda interactuar con el emisor y con su aplicación interactiva.
- **Medio de difusión:** Que está compuesto por el canal de transmisión(Emisor – Receptor) y el canal de retorno(Receptor – Emisor) que permite la comunicación entre sus actores bidireccionalmente.

Esta clasificación general se aprecia de mejor manera en la figura 2-3.

Figura 2-3: Visión general del Sistema de Televisión Digital



Fuente: GOMES, A. S y otros, Desenvolvimento de Objetos de Aprendizagem para TVDi.

2.3 SISTEMA BRASILEÑO DE TELEVISIÓN DIGITAL

2.3.1 ARQUITECTURA DEL SBTVD

El SBTVD normaliza aspectos como: el canal de interactividad, el middleware, técnicas de compresión, multiplexación, modulación, transmisión y recepción de video, audio y datos.

El Sistema Brasileño de TV Digital posee una arquitectura en capas, donde cada una de ellas proporciona servicios a la capa superior y usa los servicios que le ofrece la capa inferior como se puede observar en la figura 2-4.

- **Aplicación:** capa responsable de la captura y presentación de señales de vídeo y audio, así como de la implementación de servicios interactivos para ser

ejecutadas en el hardware(Set Top Box).

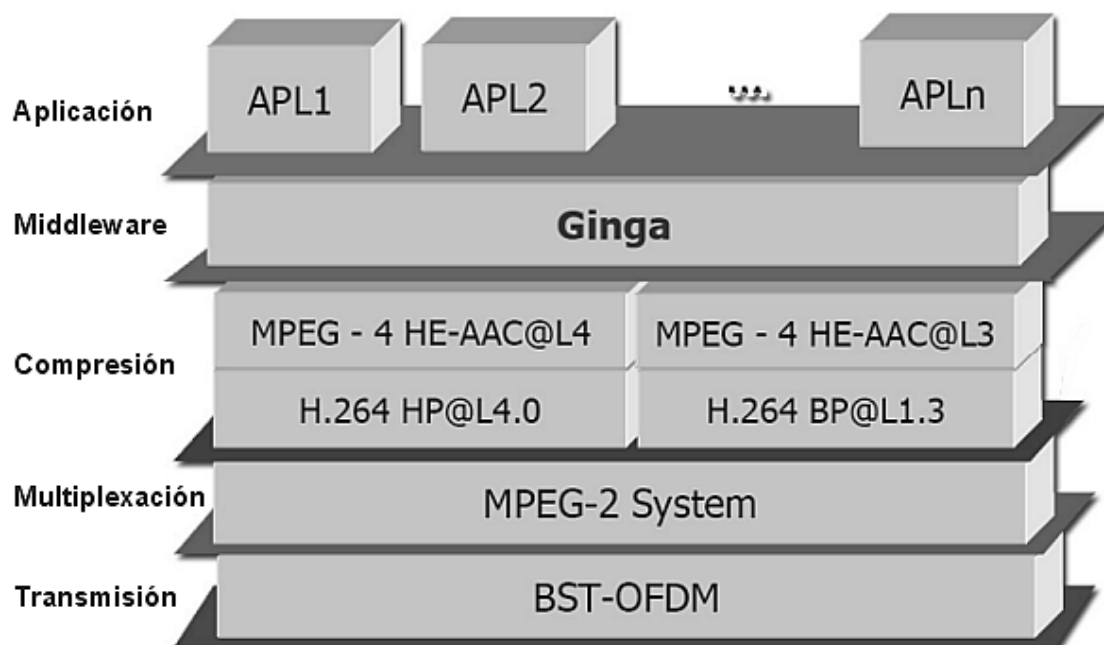
- **Middleware:** capa de software que realiza la integración de todas las subcapas del sistema. El middleware permite que las aplicaciones generadas por las estaciones sean compatibles e independientes de las plataformas³⁴ y del tipo de receptor³⁵. El middleware fue desarrollado en Brasil y se llama Ginga. Desde su nacimiento este middleware fue considerado la herramienta perfecta para la inclusión social y digital de toda la población, haciendo posible el acceso libre al conocimiento, educación a distancia, servicios sociales, etc.
- **Compresión:** capa responsable de la eliminación de redundancias en las señales de vídeo y audio, menorando así la tasa de bits necesaria para transmitir esta información. En el receptor, esta capa decodifica los flujos recibidos de audio y vídeo. El SBTVD usa la técnica de compresión H.264 o MPEG-4 AVC, que al conservar la calidad de la imagen con una tasa de datos menor a su antecesor permite hacer un uso más eficiente del espectro. Para la compresión de audio el SBTVD hace uso de la técnica de compresión MPEG-2: AAC, así como también MPEG-4: AAC que hace uso de un sistema SBR³⁶ para la alta eficiencia de codificación de audio tanto para receptores fijos como portátiles.
- **Multiplexación:** capa responsable de generar un único flujo de datos que contienen vídeo, audio y las aplicaciones interactivas de los diversos programas que serán transmitidos, basado en el estándar MPEG-2 Systems.
- **Transporte:** también llamada capa física y es la capa responsable de llevar la información digital desde la estación televisiva hasta la casa del espectador. Al igual que los demás estándares de televisión digital usa el formato de difusión Transport Stream, para la modulación el estándar BST-OFDM forma parte de la norma para menorar la pérdida de información a causa de las interferencias.

34 Se conoce como plataformas a los distintos elementos de hardware de diferentes fabricantes.

35 Los receptores pueden ser Set Top Boxes, TV, celulares, PDAs, etc.

36 Spectral band replication es una tecnología empleada para mejorar los códecs de audio y los codecs de voz, especialmente a bajos bit rate.

Figura 2-4: Arquitectura en capas de la Televisión Digital Terrestre



Fuente: SOUZA, G; SOARES, L, Interactive Television in Brazil: System Software and the Digital Divide

2.4 EL MIDDLEWARE GINGA

Un middleware para aplicaciones de televisión digital consiste principalmente en un conjunto de librerías, métodos y funciones que permiten desarrollar aplicaciones de manera fácil y rápida.

Ginga es el middleware de desarrollo de aplicaciones interactivas del estándar SBTVD. Su nombre se basa en un término usado en un fundamental movimiento de capoeira³⁷. Ginga es una especificación abierta liberada bajo licencia GPLv2³⁸ libre de regalías³⁹, brindando la posibilidad a cualquier persona de crear contenido interactivo.

³⁷ La capoeira es una expresión cultural de lucha de origen afrobrasileño practicada en Brasil, ginga es la posición básica desde la que se practican la totalidad de los demás movimientos.

³⁸ Licencia Pública General (General Public Licence) es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

³⁹ Al tener una licencia GPL el uso de Ginga es libre de pago de derechos de autor, patentes, marcas.

En definitiva, este componente es el responsable de ejecutar las aplicaciones interactivas que son enviadas en el flujo de difusión de la empresa de televisión.

2.4.1 AMBIENTES DE PROGRAMACIÓN

Las aplicaciones de televisión digital se pueden encontrar en aplicaciones declarativas o aplicaciones imperativas, dependiendo del lenguaje de programación en que estén codificadas, también se pueden construir aplicaciones híbridas, con parte de su código declarativo y parte de su código imperativo.

- **Ambientes declarativos:** En un ambiente declarativo los lenguajes de programación se especializan en describir el problema y detallar su solución a alguna tarea, sin especificar exactamente como hacerlo, por esta razón son mucho más fáciles de ser concebidos y entendidos, lo que exime la presencia de un programador especialista. Ginga da soporte para aplicaciones en el ámbito declarativo mediante el lenguaje NCL⁴⁰ que puede hacer uso de contenido en script mediante el lenguaje Lua⁴¹. Dentro del estándar este subsistema lógico toma el nombre de Ginga-NCL.
- **Ambientes imperativos:** En un ambiente imperativo los lenguajes de programación describen un conjunto de instrucciones que se ejecutan paso a paso, es decir mediante el desarrollo de un algoritmo que cambia el estado del programa para permitir encontrar su solución, esto al contrario de los ambientes declarativos exige la presencia de un programador especialista. Para aplicaciones en el ámbito imperativo el soporte viene dado mediante el lenguaje Java y toma el nombre de Ginga-J.
- **Ambientes híbridos:** En el ambiente híbrido las aplicaciones puedan ser escritas en NCL, Lua y Java, ya que, en ciertos casos para la Televisión Digital se necesitarán que se desarrolle aplicaciones interactivas con los paradigmas declarativos e imperativos, mediante las APIs⁴² contenidas en el middleware se

40 Nested Context Language

41 Lenguaje de programación imperativo, estructurado y bastante ligero, creado en el año 1993 por la Pontificia Universidad de Río de Janeiro.

42 Application Programming Interface

logra juntar ambos entornos, para lograrlo existe el subsistema lógico Ginga-CC. La arquitectura de referencia de Ginga se puede apreciar en la figura 2-5.

Figura 2-5: Arquitectura Ginga



Fuente: ABNT, Ginga-NCL para receptores fijos y móviles – Lenguaje de aplicación XML para codificación de aplicaciones, 2007

2.4.1.1 Ginga-NCL

Ginga-NCL es el ambiente de middleware Ginga responsable del procesamiento de las aplicaciones declarativas que utilizan el lenguaje NCL. Esta especificación está basada en el estándar XML⁴³, puede usar el lenguaje script Lua para aprovechar las propiedades del paradigma imperativo y ampliar las capacidades de la aplicación.

NCL define la forma en que los objetos de media⁴⁴ están organizados en tiempo y espacio, lo que aporta un mayor control sobre el contenido. Los documentos pueden importar los elementos declarados en otros documentos. NCL también es una mejor opción en el desarrollo de aplicaciones no lineales ya que es un lenguaje para la integración de los objetos de media, es decir, facilita la sincronización en tiempo y espacio, haciendo prescindibles muchas de las veces

⁴³ eXtensible Markup Language

⁴⁴ Objetos de media dentro del middleware se considera a los archivos de imágenes, video, audio, archivos Lua, archivos Java, etc.

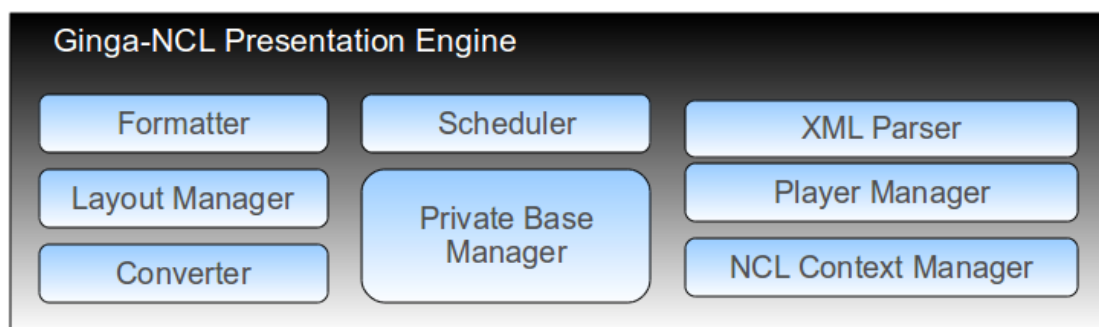
del uso de estrategias de programación o la implementación de algoritmos.

Ginga-NCL separa el contenido de la estructura de presentación, en definitiva los lenguajes declarativos permiten al desarrollador proporcionar en conjunto las tareas a realizar sin definir una gran cantidad de código para hacer las mismas tareas, en comparación con las implementaciones algorítmicas, a fin de cuentas, los lenguajes declarativos centran su uso en la adaptabilidad, edición y producción de contenido y el soporte a múltiples dispositivos.

Cuando se necesita ampliar las capacidades de la aplicación con el uso de lenguajes de script, Lua puede simplificar las tareas de manipulación de variables de forma rápida y ligera, lo que sería realmente difícil si únicamente se hiciera uso del lenguaje NCL.

En la figura 2-6 se detalla los componentes más importantes del subsistema Ginga-NCL que son:

Figura 2-6: Subsistema Ginga-NCL



Fuente: DAMASCENO, Jean Ribeiro, Middleware Ginga, 2009

- **Formatter (Formateador):** Recibe de Ginga-CC el documento NCL y se encarga de controlar su presentación, intentando garantizar que las relaciones entre los objetos de media especificados por el autor sean correctos.
- **XML Parser (Analizador XML), Converter (Convertidor):** se encarga de

realizar la traducción de la aplicación NCL en la estructura interna de datos que Ginga-NCL maneja.

- **Scheduler (Programador):** se encarga de ordenar los documentos NCL que tienen que ser cargados basados en la programación que indique que objetos de media se iniciarán dependiendo del flujo de la aplicación.
- **Player Manager (Administrador de reproducción):** recibe del programador la orden en el momento indicado para la exhibición del tipo de contenido de la media solicitada.
- **Private Base Manager (Administrador de Base Privada):** se encarga de recibir los comandos de edición de los documentos NCL y de darle mantenimiento a los documentos NCL presentados.
- **Layout Manager (Administrador de Diseño):** es el responsable de mapear todas las regiones definidas en un documento NCL.

2.4.1.2 Ginga-J

Los lenguajes imperativos se caracterizan por ofrecer mayor control del código a los desarrolladores, además de ser menos abstractos en comparación con los lenguajes declarativos. De este modo la aplicación puede ser capaz de manejar todo el control y ejecución del flujo de la aplicación. Sin embargo, el coste de diseño se incrementa en gran medida debido a la complejidad de su código fuente y no son la mejor opción al momento de desarrollar aplicaciones no lineales. Ginga-J es el responsable del procesamiento de aplicaciones procedurales escritas en lenguaje Java, que a su vez son llamadas Xlets⁴⁵. Estas aplicaciones tienen una interfaz que permite que un agente externo las inicie, pare y controle su ejecución de varias formas. Además se dividen en tres componentes: Máquina Virtual Java, el núcleo, y sus APIs.

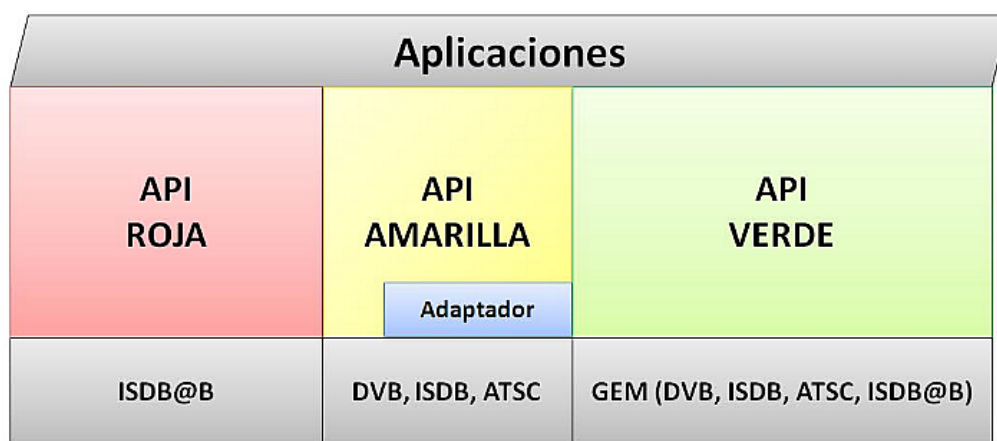
⁴⁵ El concepto Xlet es similar al de Java applet y está dirigido al desarrollo de aplicaciones para televisión digital

La implementación que Sun⁴⁶ propone a través de su lenguaje de programación Java para las aplicaciones de Televisión Digital se llama Java-DTV, que posee las siguientes funcionalidades.

- Streaming de audio y video;
- Acceso a los datos en los canales de transmisión;
- Acceso a los datos en el servicio de información;
- Control de sintonizador de canales;
- Sincronización de objetos de media; y
- Administración del ciclo de vida de la aplicación.

El API se divide en tres categorías, como se aprecia en la figura 2-7:

Figura 2-7: Subsistema Ginga-J



Fuente: SOUZA FILHO, Guido Lemos y otros. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System

- **API Verde:** mantiene compatibilidad con el estándar americano y europeo.

⁴⁶ Tras su adquisición, toda su tecnología pasó a manos de Oracle

Esta API está compuesto por los paquetes Sun JavaTV, DAVIC⁴⁷, HAVi y DVB.

- **API Amarilla:** proporciona soporte a múltiples usuarios, dispositivos y redes. Esta API está compuesto por el API JMF 2.1, el cual es necesario para desarrollar aplicaciones avanzadas, con captura de sonido al instante, con extensión para API de Presentación GEM, con funcionalidades de soporte para las especificaciones de video stream, el canal de retorno, que permite el envío asíncronico de mensajes, y extensión para los Servicios de Información.
- **API Roja:** proporciona soporte a aplicaciones que pueden ser recibidas, almacenadas y ejecutadas.

2.4.1.3 Ginga-CC

Ginga Common Core, es el subsistema lógico que provee toda funcionalidad común al soporte de los ambientes de programación declarativos, GINGA-NCL, e imperativo, GINGA-J. La arquitectura del sistema permite que únicamente el módulo Ginga-CC deba ser adaptado a la plataforma donde será implementado.

Ginga-CC provee un nivel de abstracción de la plataforma de hardware y del sistema operativo, accesible a través de las APIs. Esto le permite interactuar con el acceso al sintonizador de canal, con el sistema de archivos, el terminal gráfico entre otros.

En la figura 2-8 se puede ver los diferentes componentes que conforman el Gina-CC

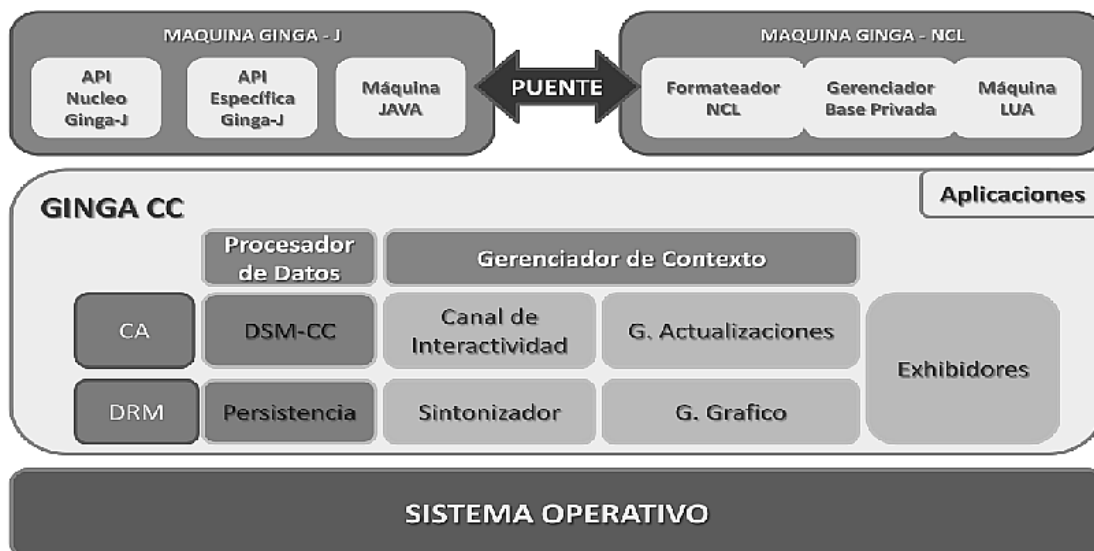
- **Sintonizador:** Este módulo es el responsable de “sintonizar” un canal, seleccionando un canal físico dentro de los flujos de transporte que están siendo enviados por ese canal.
- **Filtro de secciones:** Una vez sintonizado, el middleware debe ser capaz de

⁴⁷ Digital Audio Visual Council. Fue una asociación que acogía a diversas empresas de la industria audiovisual, el estándar ISO-IEC 16500 es una de las normas más importantes en el campo de las aplicaciones interactivas y sus servicios.

acceder a partes específicas del flujo de transporte. Para lograr este objetivo, existe el Filtro de Secciones, que es capaz de buscar en un flujo la parte exacta que las APIs necesitan para su ejecución, de ésta manera solo permite el paso de la información requerida por la API.

- **Procesador de datos:** Es el elemento responsable de acceder, procesar y reenviar los datos recibidos por la capa física. También es el encargado de notificar a los otros componentes, sobre cualquier evento que se reciba.

Figura 2-8: Subsistema GINGA-CC



Fuente: Instituto Nacional de Investigación y Capacitación de Telecomunicaciones, Investigación del Estudio del Middleware GINGA, 2010

- **Persistencia:** El middleware tiene la capacidad de almacenar y guardar archivos una vez finalizado el proceso para que luego pueda ser abierto o utilizado posteriormente.
- **Iniciador de aplicaciones:** El módulo responsable de gestionar las aplicaciones, cargar, configurar e inicializar y ejecutar cualquier aplicación tanto declarativa como imperativa. Controla el ciclo de vida de las aplicaciones, terminándolas cuando sea necesario además de controlar los recursos utilizados por las APIs.

- **Adaptador de audio video principal:** Con el adaptador de audio y video, las aplicaciones consiguen entregar un flujo de audio y video. Esto se hace necesario cuando una aplicación necesita controlar sus acciones, de acuerdo a como están siendo transmitida.
- **Gestor de gráficos:** Los estándares del middleware definen como deben ser mostrados al usuario las imágenes, los videos, los datos, etc. Gestionando las presentaciones.
- **Gestor de actualizaciones:** Componente que se encarga de gestionar las actualizaciones del sistema, verificando, y bajando las actualizaciones del middleware cuando sea necesario, para corregir los posibles errores de las versiones, ésta tarea debe realizarse en tiempo de ejecución sin incomodar el uso normal de la TV por el usuario.
- **Visualizador de medios:** Proporciona las herramientas necesarias para visualizar los archivos multimedia recibidos, como por ejemplo archivos tipo JPEG⁴⁸, MPEG⁴⁹, MP3⁵⁰, HTML⁵¹, etc.

2.5 USANDO LA INTERACTIVIDAD

El poder interactuar con las aplicaciones en un televisor con el control remoto que se lo usaba para hacer ciertas funciones limitadas, ha hecho que dentro de este contexto se establezca las directrices para satisfacer de mejor manera la experiencia del usuario con esta nueva tecnología.

Existen dos contextos claramente definidos, por un lado los usuarios que al manejar una tecnología nueva y de calidad debe sentirse complacidos de

48 Joint Photographic Experts Group, es el nombre de un comité de expertos que creó un estándar de compresión y codificación de archivos de imágenes fijas.

49 Moving Picture Experts Group, es un grupo de trabajo del ISO/IEC encargado de desarrollar estándares de codificación de audio y vídeo.

50 MPEG-1 Audio Layer III o MPEG-2 Audio Layer III, más comúnmente conocido como MP3, es un formato de compresión de audio digital.

51 HyperText Markup Language es el lenguaje de marcado predominante para la elaboración de páginas web

encontrarse con una experiencia de fácil uso, fácil aprendizaje, sin errores. En cambio para los desarrolladores también se puede aguardar ventajas importantes con respecto a la reducción de costes de soporte, motivando su uso y por ende aportando al mercado laboral trayendo consigo beneficios de comercialización.

Es prioritario que se lleve a cabo una unificación del criterio sobre el estilo de las aplicaciones interactivas de la TVD, además de participar activamente en el diseño de patrones para dispositivos de interacción como controles remoto, teclados virtuales, celulares, PDA, etc.

2.5.1 RESTRICCIONES Y DIFERENCIAS CON UN COMPUTADOR

Las aplicaciones interactivas para televisión digital no pueden ser las mismas que en su contexto web o de escritorio, esta es la principal observación que se hace a los nuevos desarrolladores de productos de software para TVD y que suele suponer un gran reto en la medida que se desprendan de las ideas que hasta ahora conocían en ambientes de programación diferentes, detalles importantes que deben tomarse en cuenta, sobre todo aquellos donde no se tiene la misma resolución que el monitor de un computador, áreas de cobertura, distante donde la señal puede sufrir algún tipo de distorsión, sin desplazamiento horizontal, hardware limitado de ciertos dispositivos, etc. A continuación en la Tabla 2-1, se detalla las principales diferencias técnicas a considerar entre un computador y un televisor. Estas características implican un enorme impacto a la hora de diseñar aplicaciones para estas plataformas.

2.5.2 ESTÁNDARES ADOPTADOS PARA LA TELEVISIÓN DIGITAL TERRESTRE

A continuación se establecen las normas básicas que permitirán desarrollar aplicaciones interactivas de mejor manera, más intuitivas, prácticas y fáciles de usar, con los colores correctos, aprovechamiento de la pantalla, navegación, textos, etc.

Tabla 2-1: Diferencias Televisor y Computador

Característica	Televisor	Computador
Resolución de la pantalla	Pequeña (640 x 480)	Mediana y grande (800 X 600 y 1280 X 1024)
Dispositivos de entrada	Control Remoto, en el mejor de los casos teclado virtual	Ratón y teclado en una posición fija
Distancia de visualización	En metros	En centímetros
Postura de usuario	Reclinado	Sentado
Ambiente	Sala de estar, ambientes que sugieren relajamiento	Escritorio, ambientes que sugieren trabajo
Integración	Varios programas de TV	Actividades personales y de trabajo
Número de usuarios	Por lo general, muchas personas están en la sala mientras el televisor está encendido. Uso social	Su uso es más bien de tipo individual
Usuario	Pasivo: Una estación de televisión envía la información solicitada, el usuario solo la recibe.	Activo: El usuario comanda, el computador es su herramienta

2.5.2.1 Tamaño de la pantalla

En la mayoría de los televisores la relación de aspecto⁵² de los televisores tradicionales es de 4:3, aunque la tendencia actual apunte a los televisores más rectangulares que tienen una relación de aspecto de 16:9. Con el afán de ocultar estas diferencias entre estos dos formatos, normalmente se utiliza un borde alrededor del contenido o incluso se apela al corte de la imagen original.

En cuanto al despliegue de una aplicación, se puede sobreponer al contenido televisivo o cambiar su tamaño. Si se va a reemplazar el video con la aplicación, esta debe ocupar los bordes de la pantalla y establecer un nivel de transparencia a su fondo. Por otro lado, si se escoge redimensionar el video, el tamaño del

⁵² La relación de aspecto (Aspect Ratio) de una imagen es la proporción entre su anchura y su altura. Se calcula dividiendo la anchura por la altura de la imagen visible en pantalla, y se expresa normalmente como «X:Y»

mismo no puede ser menor a la cuarta parte de la pantalla.

2.5.2.2 Colores

Entre un televisor y un monitor de computador existe una gran diferencia en su *gamute*⁵³. Por eso se recomienda no utilizar colores demasiado saturados o con su luminosidad alta, esto quiere decir que los colores deben estar entre valores del sistema RGB⁵⁴, no menor a 16 unidades y no mayor a 236 unidades en la escala de 0 a 256.

2.5.2.3 Texto

En el manejo del texto hay que ser más cuidadoso, ya que, los usuarios no están acostumbrados a leer demasiado en las pantallas de sus televisores, además de que adaptar el texto a la gran diversidad de dispositivos y televisores que existen es un verdadero desafío. Por esta razón y debido a estas limitaciones se hace recomendable usar un tamaño de texto no menor a 18 puntos, se aconseja que sea de 22 puntos o más. Con el afán de aumentar la legibilidad. Se propone estos valores para las aplicaciones:

- Títulos: 36 pt
- Menús: 20 pt
- Texto diverso: 22 pt
- Botones: 18 pt

El tamaño máximo de letras en toda la pantalla no debe de pasar de 90. Usar el texto con colores claros sobre un fondo oscuro también permitirá una mejor lectura.

⁵³ Conjunto de colores que soporta un dispositivo.

⁵⁴ Red, Green, Blue; hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul.

2.5.2.4 Dispositivo de interacción

El dispositivo de interacción es de la misma importancia que la de un televisor, el no establecer una norma para el diseño de los controles de acceso puede conllevar a una dificultad en la usabilidad. De tal manera que la norma indica que se debe disponer de un control remoto con teclas dedicadas a las funcionalidades específicas, este evita que el usuario se canse con los inesperados eventos generados por un confuso acceso a la aplicación interactiva.

2.5.2.5 Navegación

En cuanto a la navegación existen dos posibilidades que se presentan. Primero se puede hacer uso de flechas de dirección más una tecla de confirmación o se puede usar flechas de dirección sin una tecla de confirmación. El cursor utilizado debe ser prominente y claramente separado del resto de la interfaz, a esto sumado el cuidado que se debe tener para saber cuando algo esta seleccionado o no lo está.

2.6 CONSIDERACIONES FINALES

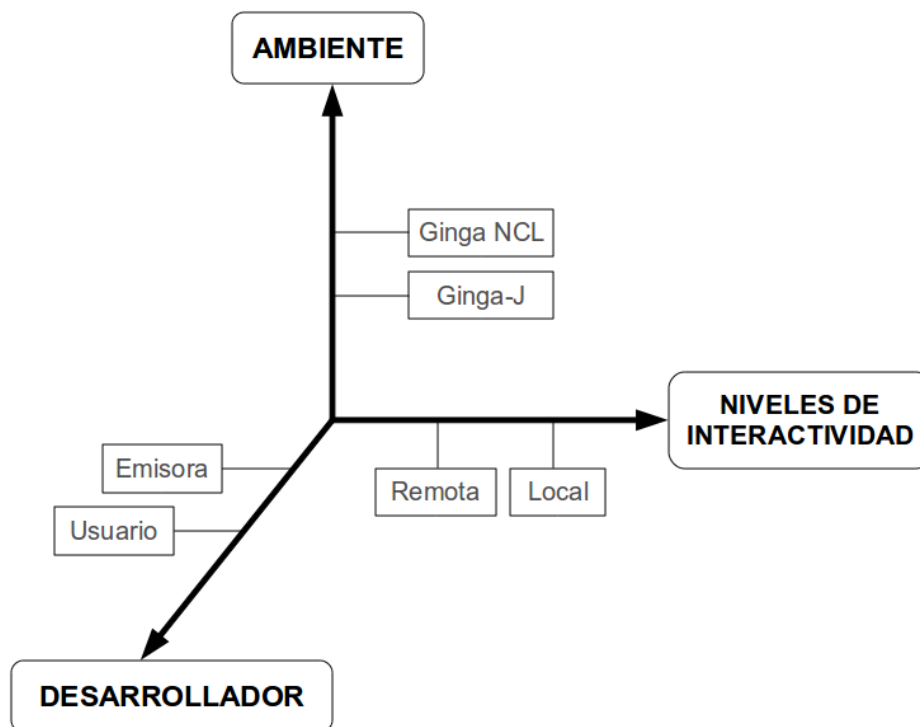
Este capítulo esta orientado a descubrir el mundo que envuelve la interactividad en la televisión digital, sobretodo para orientar, alentar y educar a los programadores interesados en desarrollar aplicaciones para TVD, tomando en cuenta las precauciones descritas anteriormente sobre su funcionamiento y presentación, ya que, muchas de las personas a las que están dirigidas estas aplicaciones no han usado antes un computador y el uso para su televisor es de lo más básico, como cambiar de canal o manipular el volumen.

CAPÍTULO 3. DISEÑO DE APLICACIONES INTERACTIVAS

La televisión digital es más que imagen y sonido de alta definición. Con ella se puede desarrollar para televisión, aplicaciones con capacidad computacional, como las aplicaciones de escritorio, que permiten almacenar y procesar la información, y con la ayuda del canal de retorno permite la construcción de aplicaciones interactivas.

Las opciones de desarrollo se pueden representar en tres dimensiones como se observa en la figura 3-1:

Figura 3-1: Dimensiones en el desarrollo de aplicaciones interactivas



Fuente: ALBUQUERQUE, Antônio; LOPES, João. A Tv Digital no Brasil e o desenvolvimento de aplicações interativas para o middleware Ginga. 2008

- **Desarrollador (Estación televisiva o usuario):** Las aplicaciones pueden ser desarrolladas por personal especializado en un trabajo exclusivo para las televisoras o también desarrollada por los usuarios o programadores

aficionados sin ninguna relación profesional con las estaciones de televisión.

- **Niveles de interactividad (Local o remota):** En el caso que la aplicación sea desarrollada por la estación de televisión, la aplicación será enviada al usuario a través del canal de difusión al receptor o Set Top Box, en este caso el nivel de interactividad es remota. En cambio si la aplicación es desarrollada por el usuario esta será enviada al receptor o Set Top Box mediante una entrada externa, como un puerto USB, un puerto de red o una tarjeta de memoria, donde se considerará a la interactividad como local.
- **Ambiente de ejecución:** Como se ha visto en el capítulo anterior, Ginga ofrece dos entornos de ejecución: el entorno de las aplicaciones declarativas Ginga-NCL y el entorno para aplicaciones procedurales Ginga-J.

Las aplicaciones que se desarrollen para el sistema de televisión digital pueden ser divididos en un conjunto de aplicaciones declarativas y un conjunto de aplicaciones procedurales. El lenguaje declarativo se centra mucho más en un área u objeto específico, no puede centrarse en que este tipo de lenguaje sea empleado de manera general para todos los casos. Cuando el foco del problema no coincide con el foco del lenguaje, su resolución, aunque no imposible puede resultar demasiado compleja. En este caso se recomienda el uso de un lenguaje procedural.

Una vez que el desarrollador defina el nivel de interactividad y el ambiente de ejecución, se puede establecer los siguientes pasos para empezar a crear aplicaciones interactivas. El diagrama que se muestra en la figura 3-2 representa las opciones de desarrollo disponibles en cada entorno.

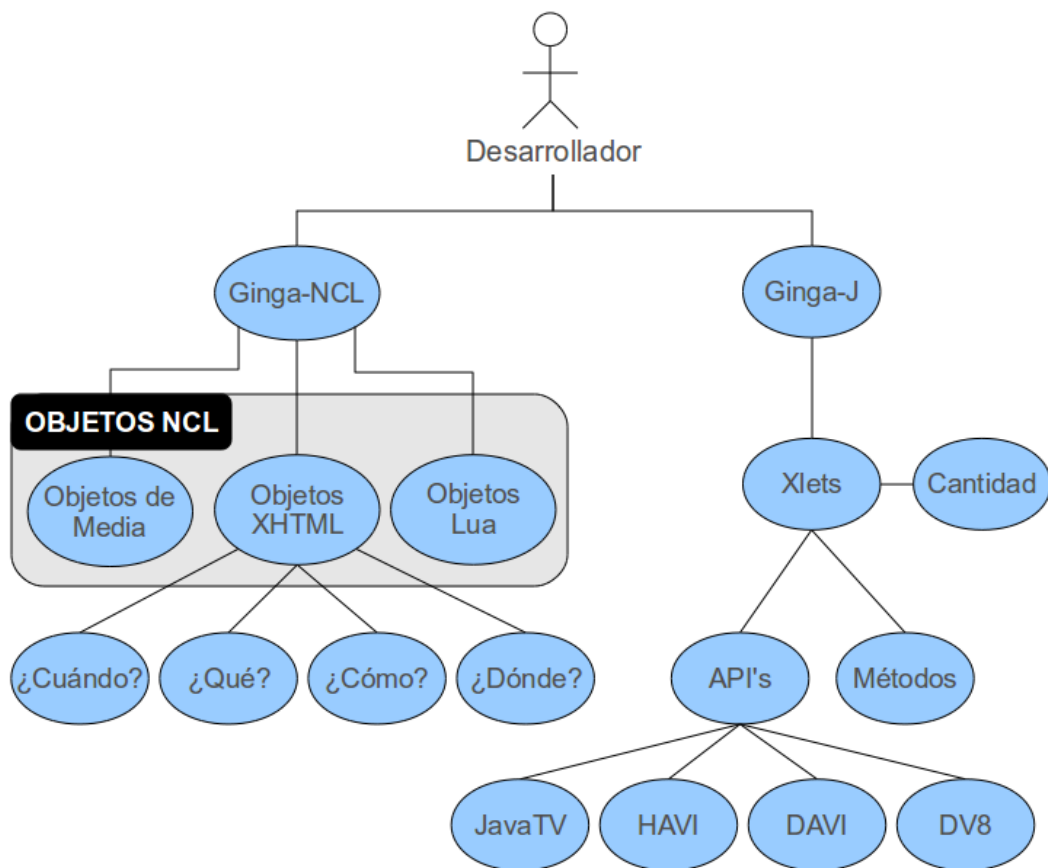
En ambientes Ginga-NCL se puede crear objetos que se mostrarán a través del formateador NCL. Estos objetos también son llamados como “nodo de contenido” y pueden ser:

- Un objeto de media que está asociado a un objeto multimedia como video,

audio, imagen, texto, etc.

- Un objeto XHTML que está asociado a un elemento XHTML.
- Un objeto Lua que está asociado a un elemento que da soporte a programación procedural mediante scripts Lua.

Figura 3-2: Diagrama con opciones de desarrollo



Fuente: ALBUQUERQUE, Antônio; LOPES, João. A Tv Digital no Brasil e o desenvolvimento de aplicações interativas para o middleware Ginga. 2008

En el entorno de ejecución de Xlets, el Ginga-J, se define la cantidad de Xlets necesarias. La cantidad depende del número de distintas características que se llevarán a cabo para resolver el problema. Se recomienda crear una Xlet para cada función específica como se ve en la figura 3-3

Figura 3-3: Pantalla mostrando varias Xlets



Fuente: ALBUQUERQUE, Antônio; LOPES, João. A Tv Digital no Brasil e o desenvolvimento de aplicações interativas para o middleware Ginga. 2008

3.1 APLICACIONES INTERACTIVAS EN AMBIENTES GINGA-NCL

3.1.1 NCL

NCL es un lenguaje basado en XML, NCL se utiliza para recoger objetos en una presentación multimedia. Estos objetos pueden ser de cualquier tipo, audio, video, imagen, HTML o incluso un objeto procedural. NCL se basa en el modelo NCM (Nested Context Model).

3.1.1.1 Nested Context Model

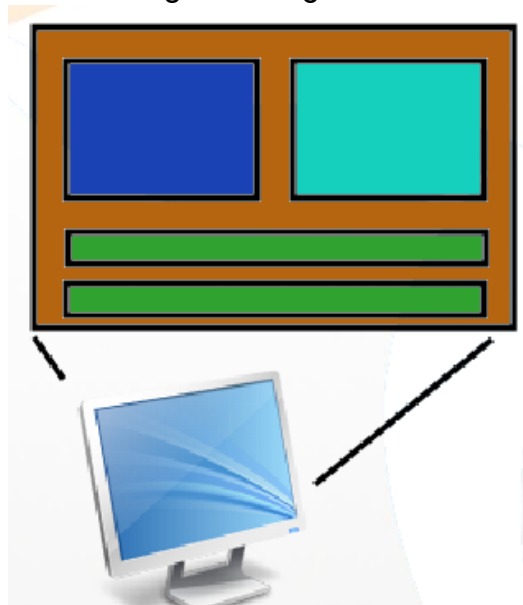
Un documento hipermedia, en forma genérica, se compone de nodos y enlaces. Los nodos representan abstracciones de los medios utilizados en el documento, también proporciona información adicional, acerca de su presentación. Los enlaces son la sincronización espacial o temporal entre los nodos que componen el documento.

El modelo NCM (Nested Context Model), Modelo de Contextos Anidados, se extiende por encima de los conceptos mediante el aumento de la potencia y flexibilidad de un documento hipermedia. NCM amplía la definición de dos tipos de nodos, nodos de contenido y nodos de composición, un nodo contiene información acerca de una media utilizada por el documento, mientras que, un nodo compuesto tiene un conjunto de nodos de contenido y/u otros nodos de composición y el número de enlaces que se utiliza para dar forma y organización a un documento hipermedia.

Todo objeto debe establecer su comportamiento, para ello en su declaración debe indicar lo que debe ser presentado, cómo, cuándo y dónde debe ser presentado.

- **¿Dónde?** Para que un nodo de un documento hipermedia sea presentado, es necesario definir un área para la exposición de los mismos. El modelo NCM definen los elementos específicos para este fin llamadas “regiones”. Estos factores indican la posición y el tamaño del área en la que podrán presentarse, sin preocuparse por la definición de lo que realmente se muestren en este ámbito. Figura 3-4.

Figura 3-4: ¿Dónde?



Fuente: CARVALHO, Rafael y otros. Introdução às Linguagens NCL e Lua: Desenvolvendo Aplicações Interativas para TV Digital. 2009

- **¿Cómo?** La definición de una región debe ser complementado con otras informaciones que indican cómo cada nodo se mostrará. Esta descripción complementa las características de la presentación que cada nodo realiza a través de elementos llamados “descriptores”. Un descriptor puede describir los parámetros de la presentación del nodo, incluyendo la región que será presentado, su transparencia, la duración de su presentación, entre otros. Figura 3-5.

Figura 3-5: ¿Cómo?

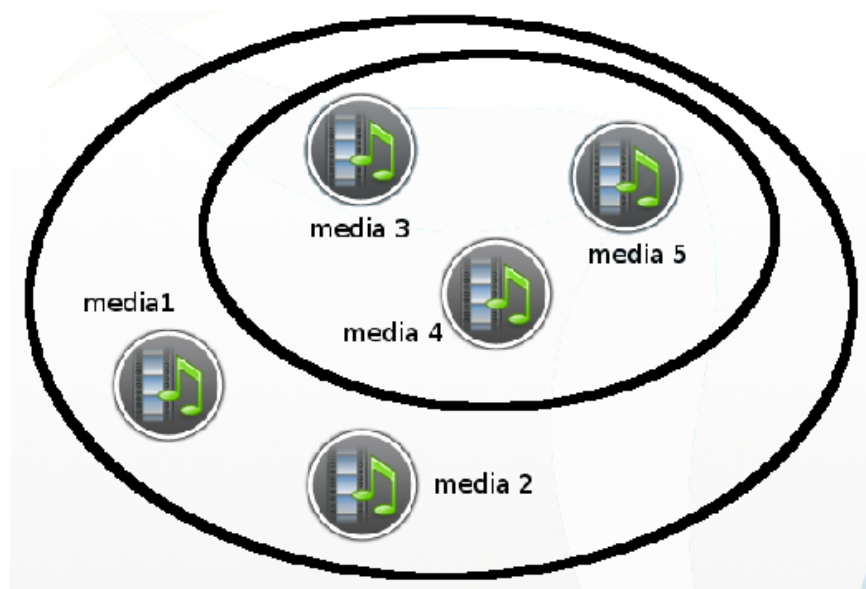


Fuente: CARVALHO, Rafael y otros. Introdução às Linguagens NCL e Lua: Desenvolvendo Aplicações Interativas para TV Digital. 2009

- **¿Qué?** El contenido de un documento hipermedia está representada por elementos llamadas medias. Una media representa cada nodo de un documento, indicando el descriptor al que se refiera. De acuerdo con NCM, una media necesariamente debe estar dentro de un nodo de composición denominado contexto, que se utiliza para representar un documento completo o parte del mismo. Figura 3-6 muestra el concepto de medias y contextos.
- **¿Cuándo?** Una vez que haya seleccionado los nodos que forman parte del documento de hipermedia, se hace necesario definir cual será el primer nodo a mostrar y el orden de presentación de los otros nodos. Esta definición se hace con el uso de elementos llamados puertos y enlaces (links). Los puertos definen los nodos que serán presentados, cuando un nodo de contexto se ha iniciado y los links definen las relaciones entre la sincronización de los nodos y la interactividad del programa. Un link, sin embargo, no define todo el comportamiento de una relación de por sí, esto requiere el uso de conectores.

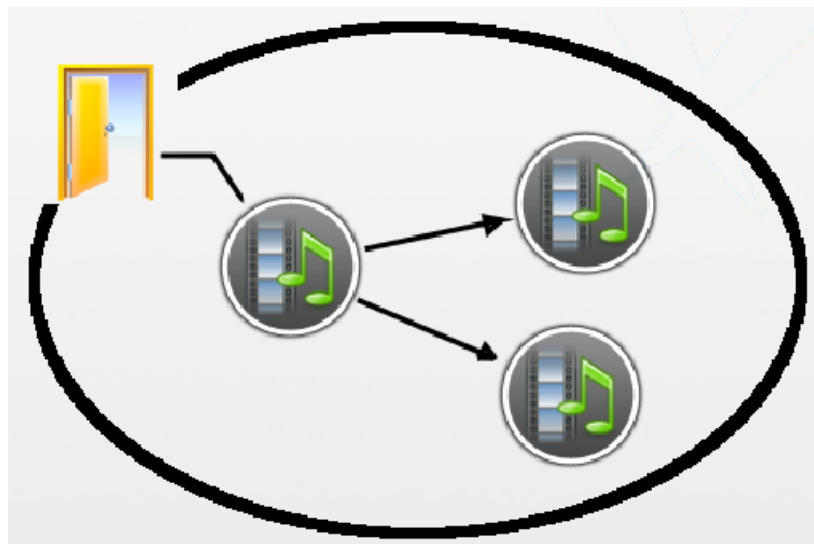
Figura 3-7 muestra el uso de los puertos y enlaces.

Figura 3-6: ¿Qué?



Fuente: CARVALHO, Rafael y otros. Introdução às Linguagens NCL e Lua: Desenvolvendo Aplicações Interativas para TV Digital. 2009

Figura 3-7: ¿Cuándo?



Fuente: CARVALHO, Rafael y otros. Introdução às Linguagens NCL e Lua: Desenvolvendo Aplicações Interativas para TV Digital. 2009

3.2 APLICACIONES INTERACTIVAS EN AMBIENTES GINGA-J

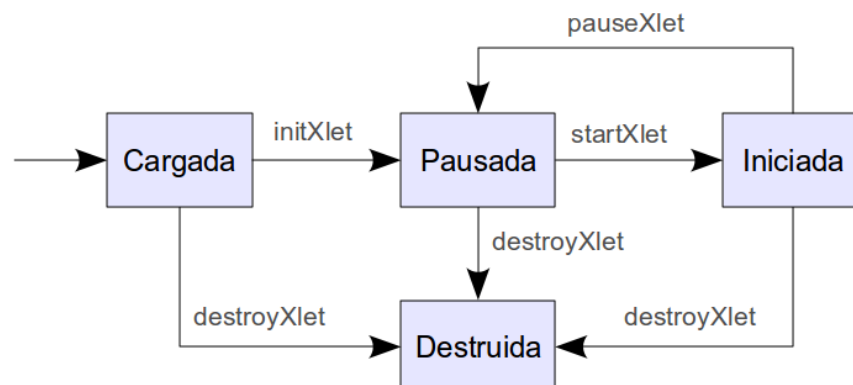
3.2.1 CICLO DE VIDA XLETS

Los *xlets* son los aplicativos para entornos de televisión, parecidos a los *applets* pero con ciertas diferencias y limitaciones debido a su funcionalidad y al hardware que lo soporta, estas aplicaciones se las puede considerar en sus distintos estados y permanecen sobre la máquina virtual mientras se ejecutan. Las *xlets* deben implementar los métodos de la interfaz *javax.microedition.xlet.Xlet* para determinar en que estado se encuentra y proceder con la transición a un nuevo estado si así se determina.

En la figura 3-8 se puede apreciar los distintos estados en que la aplicación puede encontrarse, así, un *Xlet* tendrá cuatro estados principales: Cargado, pausado, arrancado y destruido.

El gestor de aplicaciones carga el archivo con la clase principal y crea una instancia del *Xlet* llamando al constructor por defecto. Esto puede pasar en cualquier momento una vez que la aplicación haya sido indicada. En este momento, el *Xlet* alcanza el estado Cargado.

Figura 3-8: Estados de un Xlet



Fuente: ABNT NBR 15606-4 , GINGA-J. Ambiente para la ejecución de aplicaciones procedurales . 2010

El *Xlet* puede ser iniciado de tres formas:

- Por indicación directa del usuario,
- Invocado por otra *xlet*,
- Automático por configuración en la AIT (Tabla de información de las aplicaciones)

Al inicializarse el *xlet* empieza la pre carga de datos de gran tamaño como imágenes que pueden requerir tiempo para cargarse. Cuando la inicialización se ha completado, el *Xlet* está en el estado “Pausa” y puede comenzar inmediatamente.

Una vez que el método *initXlet()* ha terminado, el gestor de aplicaciones llama el método *startXlet()*. Esto pasará el *Xlet* del estado “Pausa” al estado “Arrancado” (*Started*), y el *Xlet* podrá interactuar con el usuario.

Durante la ejecución del *Xlet*, el gestor de aplicaciones puede llamar al método *pauseXlet()*. Esto hará que la aplicación pase del estado “Arrancado” al “Pausado”. La aplicación puede volver a ejecutarse de nuevo llamando al método *startXlet()*. Esto puede pasar varias veces durante la vida del *Xlet*.

Al final de su vida, el gestor de aplicaciones llamará al método *destroyXlet()*, que hará que el *Xlet* pase al estado “Destruído” (*Destroyed*) y libere todos sus recursos. En ese punto, no se puede volver a ejecutar esa instancia del *Xlet*.

Los métodos que son necesarios para cada caso en el que se encuentre el *xlet* se detallan a continuación.

- **initXlet:** Este es el método donde las variables y las instancias del objeto se inicializan. En ella el contexto *XletContext* se pasa a la *Xlet* para su registro a la espera de alguna petición de ejecución. Este método es llamado una sola vez en el ciclo de vida de *Xlet*.

- **startXlet:** Este método debe implementar todos los procedimientos necesarios para que la aplicación entre en estado de ejecución.
- **pauseXlet:** En este método se procede a detener la aplicación y sus servicios, el *xlet* entra en modo de espera.
- **destroyXlet:** Este método debe implementar todos los procedimientos correctamente para que la ejecución del *xlet* finalice de manera correcta, al terminar este proceso, se liberan los recursos de hardware, se guardan datos si es necesario, etc.

El gestor de aplicaciones se encarga de manejar los estados de cada aplicación, tener disponibles los recursos de hardware, administrar la memoria, manejar los errores de ejecución de las aplicaciones y mediante el objeto *XletContext* el gestor de aplicaciones se comunica con las *xlets* para ser notificado de sus cambios de estado.

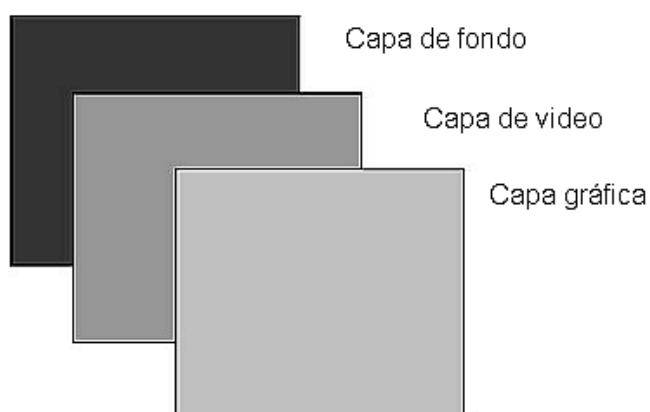
El *interface* de *XletContext* define los siguientes métodos:

- **notifyDestroyed:** Este método indica al gestor de aplicaciones que el *xlet* a entrado al estado “Destruído” después de que la aplicación a finalizado su ejecución y está listo para ser destruido.
- **notifyPaused:** Este método indica al gestor de aplicaciones que el *xlet* a entrado en el estado “Pausado”. La aplicación debe entrar en este estado luego que en un período de tiempo el *xlet* no ha entregado ningún servicio.
- **getXletProperty:** Este método permite al *Xlet* obtener información sobre el ambiente de ejecución.
- **resumeRequest:** Este método indica al gestor de aplicaciones que un *Xlet* quiere entrar en estado de ejecución.

3.3 AMBIENTE GRÁFICO

En la mayoría de ambientes de desarrollo de aplicaciones multimedia es de uso común un modelo de pantalla que gestione la manera de presentarlos al usuario.

Figura 3-9: Modelo para la presentación de las aplicaciones interactivas



Fuente: HIERA, Lucas. Análise de Middlewares no contexto de Sistemas Terrestres de Televisão Digital . 2008

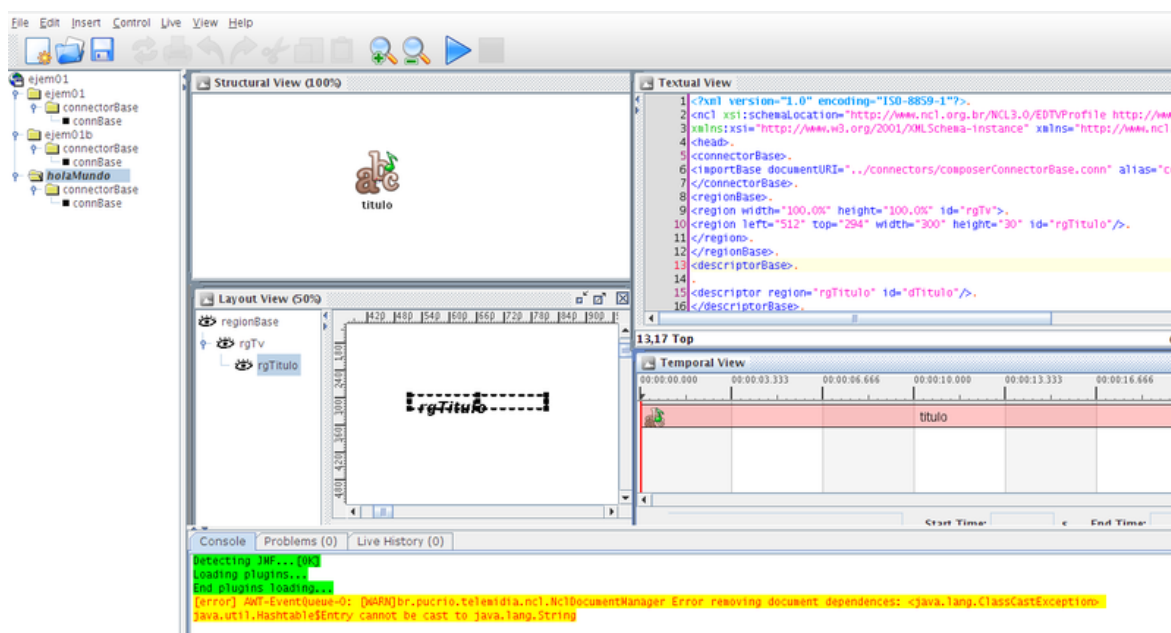
Este modelo es el que mejor se adapta a las necesidades del entorno de desarrollo de aplicaciones para televisión interactiva. Como se aprecia en la figura 3-9, tres capas son definidas para cubrir estos aspectos: una capa de imagen del fondo de la pantalla (background), una capa de vídeo y una capa gráfica que muestra la interfaz de usuario de la aplicación. Las capas son superpuestas para producir la imagen que ve el usuario final.

3.4 HERRAMIENTAS DE DESARROLLO

3.4.1 COMPOSER

Composer es una herramienta desarrollada por el Laboratorio TeleMídia del Departamento de Informática de la PUC-Rio. Con esta herramienta es posible construir programas audiovisuales interactivos con un mediano conocimiento del lenguaje NCL.

Figura 3-10: Herramienta Composer



Fuente: El autor

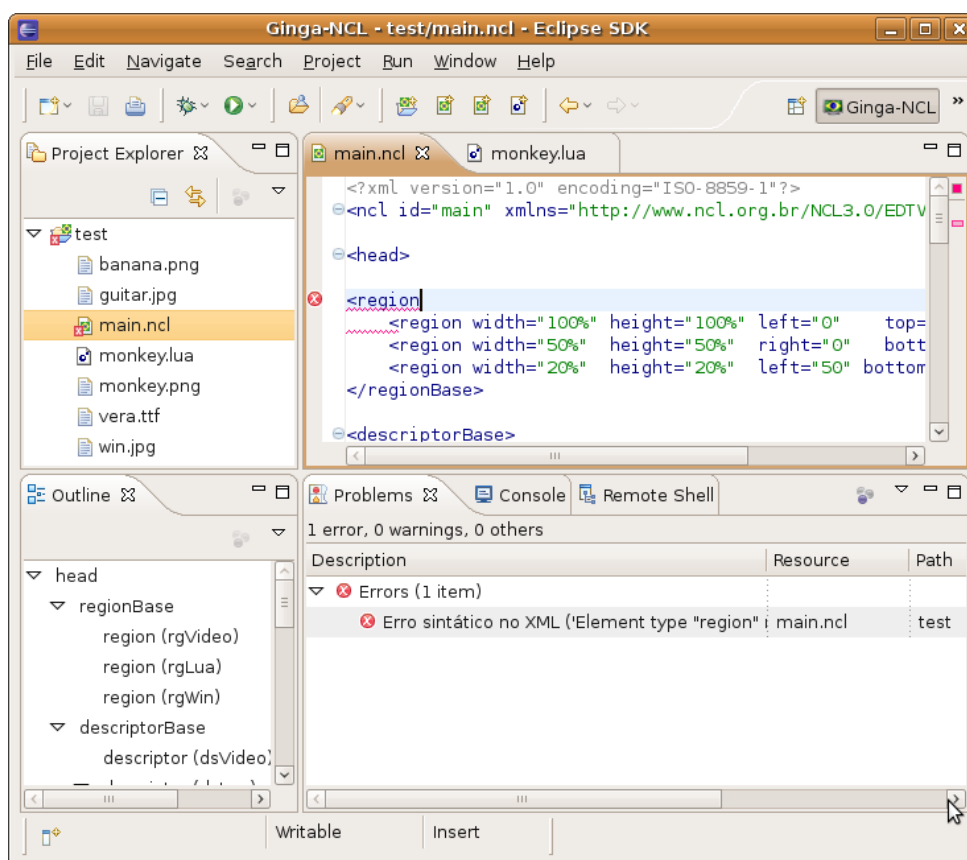
Actualmente esta herramienta se encuentra en la versión 2.2.1 NCL 3.0 y es posible encontrarla para diversos sistemas operativos de 32 bits como Microsoft Windows, GNU/Linux y Mac OS X.

La herramienta Composer es dinámica y bastante completa ya que posee una gran variedad de formas en la que se puede visualizar el proyecto. En la figura 3-10 se presenta el ambiente gráfico de trabajo de Composer.

3.4.2 GINGAWAY

Gingaway (Figura 3-11) es una herramienta especializada en el desarrollo de aplicaciones interactivas para la TV digital. Ésta es una extensión para la plataforma Eclipse que facilita la codificación de aplicaciones Ginga-NCL con soporte para scripts Lua-GingaWay se basa en el plugin de Eclipse, tales como, NCL Eclipse o Lua Eclipse, y engloba sus funcionalidades.

Figura 3-11: Herramienta Gingaway



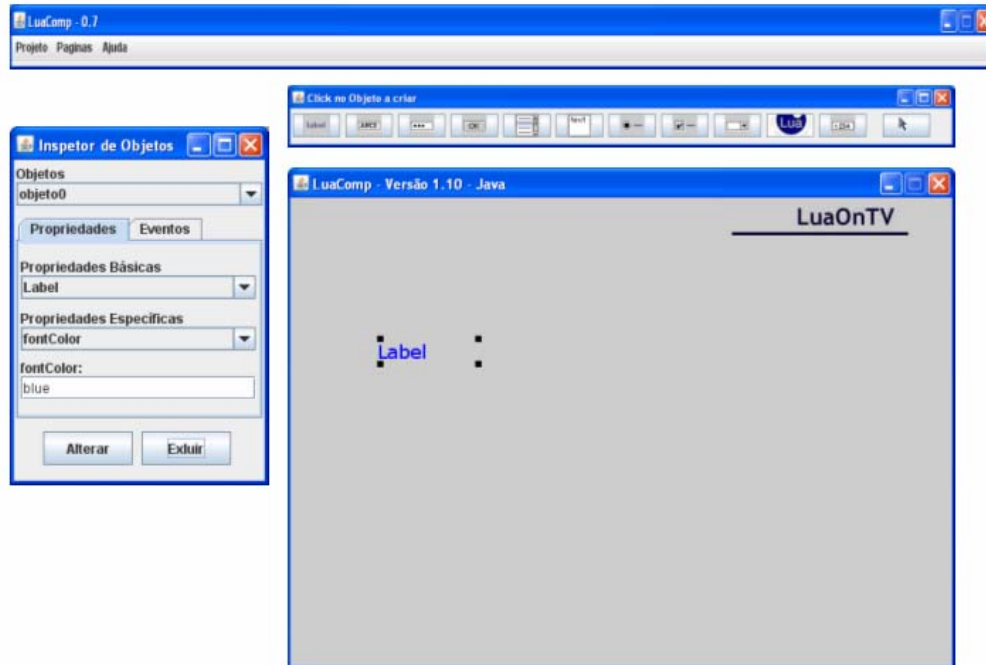
Fuente: BELTRÃO, Mauro. Gingaway, uma ferramenta para criação de aplicações ginga-ncl interativas para tv digital. 2008

3.4.3 LUACOMP

Herramienta de autoría para aplicaciones Ginga-NCLua para TV Digital interactiva. LuaComp(Figura 3-12) es una herramienta que posibilita la creación de aplicaciones fácil y rápidamente, explorando funcionalidades de LuaOnTV. Esta herramienta presenta el concepto de visores como Composer, basado en el mismo concepto.

LuaComp dispone de una interfaz gráfica para la generación de código de aplicaciones(ncl, lua y xml), se usa como principal característica de desarrollo el formato gráfico *drag and drop*(arrastrar y soltar) que consiste en seleccionar componentes pre establecidos.

Figura 3-12: Herramienta LuaComp



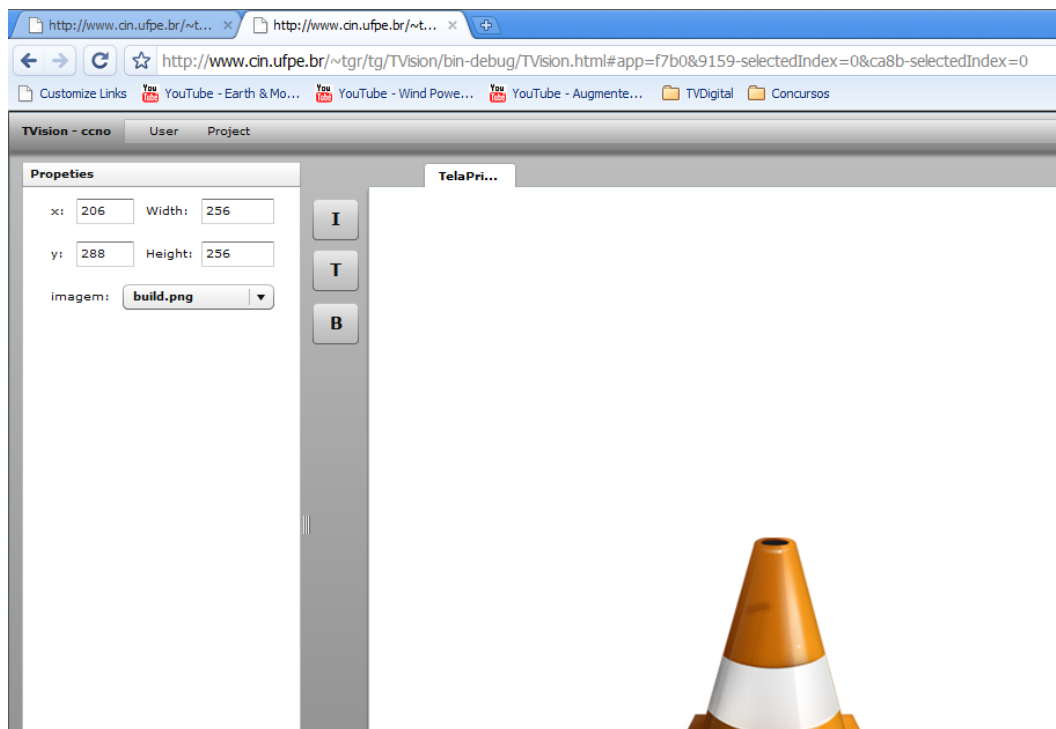
Fuente: DE SOUZA, Paulo. Luacomp: ferramenta de autoria de aplicações para tv digital. 2009

3.4.4 TVISION

TVision (Figura 3-13) es una herramienta que se ejecuta en navegadores web, utilizando tecnologías de tipo RIA (Rich Internet Application), que consigue una interfaz amigable e intuitiva. Tiene por objeto el facilitar el desarrollo de aplicaciones interactivas, haciendo uso del formato *drag and drop*. Esta herramienta genera código NCL y también tiene soporte para aplicaciones que usen scripts Lua.

TVision carece de documentación y de herramientas de pruebas o validaciones para las aplicaciones, tampoco tiene una herramienta de simulación integrada, además de no disponer de un ambiente de desarrollo *offline*, obligando al desarrollador a disponer de acceso a Internet para poder desarrollar las aplicaciones.

Figura 3-13: Herramienta Tvision



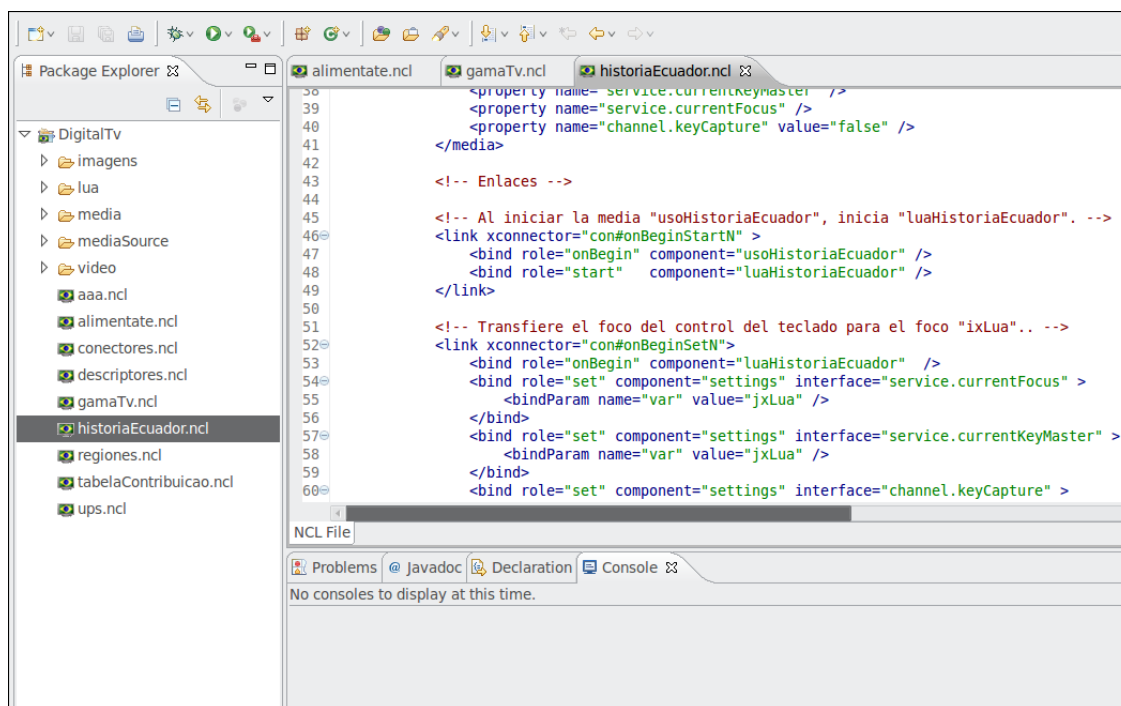
Fuente: NEVES DE OLIVEIRA, Tvision. Ferramenta Gráfica para Desenvolvimento de Aplicações Para TV Digital no Formato GINGA-NCL. 2009

3.4.5 NCL ECLIPSE

NCL Eclipse (Figura 3-14) es un plugin para el IDE Eclipse que ofrece diversas funcionalidades centradas en el desarrollo como por ejemplo: generación de código automática, validación de documentos NCL, coloración de elementos XML y de palabras reservadas del lenguaje. Eclipse es una herramienta que integra diversas necesidades, también ofrece plugins para Lua, como LuaEclipse.

Por presentar diversas facilidades para la edición de documentos NCL y por integrar eficientemente con LuaEclipse, esta herramienta usualmente se utiliza como base para las demás soluciones que buscan facilitar o desarrollo de aplicaciones declarativas.

Figura 3-14: Herramienta NCL Eclipse



Fuente: El autor

A través del análisis de estos trabajos es posible identificar algunas características así como también algunos puntos negativos que son tomados en cuenta para definir con que herramienta se desarrollan las aplicaciones interactivas. En la tabla 3-1 se evidencia los puntos positivos y los puntos negativos de cada herramienta.

Analizado todos estos factores, la herramienta para el desarrollo de la aplicación interactiva es NCL Eclipse por los siguientes detalles:

- Integración tanto de NCL así como también de Lua
- La popularidad y confiabilidad que este entorno de desarrollo tiene entre la mayoría de programadores, su documentación y comunidad a nivel mundial
- La experiencia en el uso de Eclipse en diversos proyectos de desarrollo permiten tomar la decisión por esta opción. En el Anexo A. se detalla la instalación y configuración del entorno de desarrollo escogido.

Herramienta	Puntos Positivos	Puntos negativos
Composer	<ul style="list-style-type: none"> • Edición de documentos NCL; • Varios visores; • Manipulación gráfica de componentes; • Integración con herramientas de simulación; • Generación de código 	<ul style="list-style-type: none"> • No es posible editar documentos Lua; • Emulador no soporta Lua; • No es posible integrar con herramientas de simulación; • No provee una orientación arquitectural del desarrollador;
GingaWay	<ul style="list-style-type: none"> • Edición de documentos NCL; • Edición de documentos Lua; • Integración con herramientas de simulación; • Configuración de nuevas herramientas de simulación; 	<ul style="list-style-type: none"> • No es posible manipulación gráfica de componentes; • No es posible crear componentes configurables; • No provee una orientación arquitectural del desarrollador; • Poca documentación
LuaComp	<ul style="list-style-type: none"> • Edición de documentos NCL; • Edición de documentos Lua; • Varios visores; • Manipulación gráfica de componentes; • Generación de código; • Componentes de interfaz gráfica configurable; • Integración con herramientas de simulación; • Facilita la integración con otras herramientas. • Manipulación gráfica de componentes; 	<ul style="list-style-type: none"> • Falta de calidad de los componentes; • No es posible integrar con herramientas de simulación; • No provee una orientación arquitectural del desarrollador; • Poca documentación
Tvision	<ul style="list-style-type: none"> • Edición de documentos NCL; • Edición de documentos Lua; • Varios visores; • Componentes de interfaz gráfica configurable; 	<ul style="list-style-type: none"> • Pocos componentes de interfaz gráfica disponible; • No es posible integrar con herramientas de simulación; • No provee una orientación

Herramienta	Puntos Positivos	Puntos negativos
	<ul style="list-style-type: none"> • Ambiente de desarrollo online; • Generación de código; 	<ul style="list-style-type: none"> arquitectural del desarrollador; • No existe un ambiente de desarrollo <i>offline</i>; • Poca documentación
NCL Eclipse	<ul style="list-style-type: none"> • Edición de documentos NCL; • Edición de documentos Lua; • Integración con herramientas de simulación; 	<ul style="list-style-type: none"> • Pocos componentes de interfaz gráfica disponible; • No provee una orientación arquitectural del desarrollador;

Tabla 3-1: Comparativa entre herramientas de desarrollo

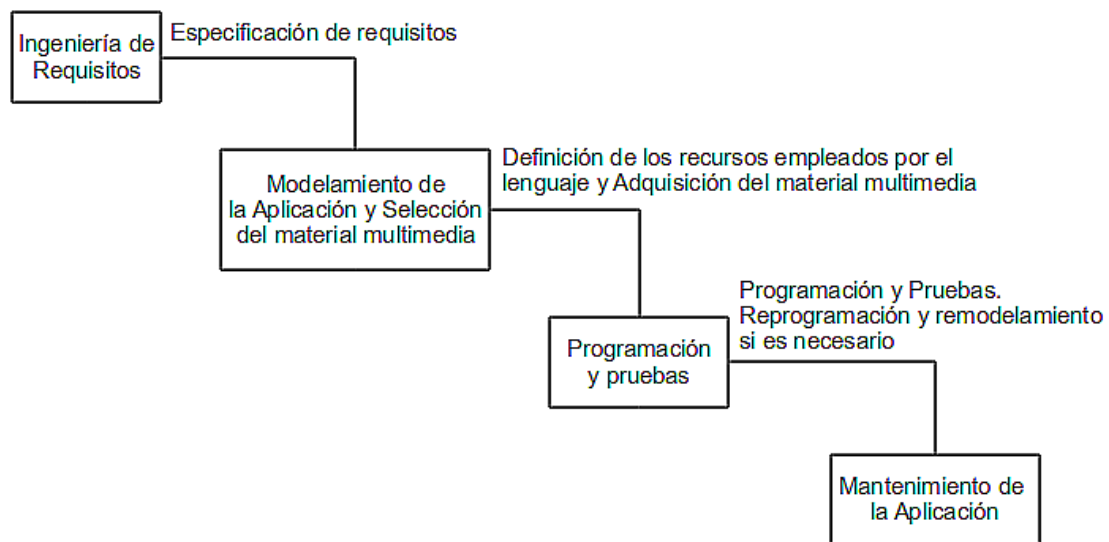
3.5 DIGITALTV

La aplicación que se desarrolló para la puesta en práctica de todos los conocimientos adquiridos en esta investigación lleva por nombre DigitalTv que abarca un conjunto de 5 aplicaciones que se pueden acceder a través de un menú principal. Las aplicaciones que forman parte de DigitalTv son:

- **Aliméntate Ecuador:** Una sencilla pero divertida aplicación para aprender sobre nuestra dieta y nuestros hábitos alimenticios, en un ámbito T-Health
- **Historia del Ecuador:** A manera de trivía se pretende entretener y educar al usuario con temas de la historia de nuestro país, en un ámbito T-Learning
- **Entidades Públicas:** Consulta la base de datos del Sistema de Actas de Finiquito del Ministerio de Relaciones Laborales para consultar los datos sobre un turno de un usuario en un ámbito T-Government.
- **U.P.S:** Información de la Universidad Politécnica Salesiana, su misión y visión.
- **GamaTV:** Mediante las redes sociales y en este caso usando twitter se puede mantener al tanto a los televidentes de toda la información acerca de esta televisora.

En términos de ingeniería de software para el desarrollo de esta aplicación se ha utilizado un ciclo de vida en cascada, ya que, es prioridad el optimizar el tiempo de desarrollo, por lo tanto, se ha optado por esta metodología. En la figura 3-15 se puede apreciar el modelo en cascada del proyecto.

Figura 3-15: Ciclo de vida de la aplicación



Fuente: El autor

Dado que la aplicación iba a generar gran cantidad de código y para mejorar la manera de leer el código y poder darle un mantenimiento efectivo sin mayor problema, la aplicación se separó en diversos archivos NCL y Lua de la siguiente manera:

- **main.ncl:** En este archivo se encuentra el menú principal, desde aquí se instancia a los demás archivos, regiones, conectores y descriptores se manejan los contextos y se lanzan cuando sea el caso las demás aplicaciones.
- **regiones.ncl:** En este archivo se organiza el “*layout*” de la aplicación, es decir, se organiza cada área de la pantalla donde los nodos de media tomarán su lugar a fin de tener la presentación visual de la aplicación.
- **conectores.ncl:** En este archivo se define los papeles (roles) que los nodos de origen y de destino ejercen en los enlaces que se van a utilizar, es decir,

aquí se define la manera en que la aplicación responde a la interacción del usuario.

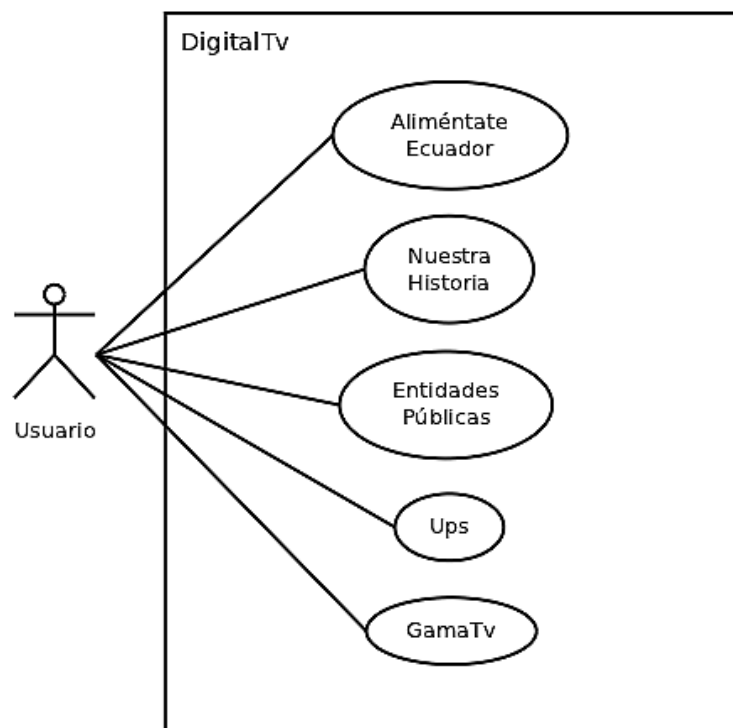
- **descriptores:** En este archivo se define como se va a presentar una región, por ejemplo, si tiene borde, si recibe el foco, etc., también se especifica las variables encargadas de manejar el paso de parámetros entre archivos NCL y archivos Lua. Este archivo esta asociado con el archivo *regiones.ncl*.
- **alimentate.ncl:** En este archivo se encuentra el código de la aplicación Aliméntate Ecuador
- **historiaEcuador.ncl:** En este archivo se encuentra el código de la aplicación Historia del Ecuador.
- **ministerios.ncl:** En este archivo se encuentra el código de la aplicación Ministerios.
- **ups.ncl:** En este archivo se encuentra el código de la aplicación Ups.
- **gamaTv.ncl:** En este archivo se encuentra el código de la aplicación GamaTv

Las aplicaciones que hacen uso del lenguaje Lua tienen sus archivos en un directorio diferente llamado lua. Dentro de este directorio se encuentran archivos como *tcp.lua*, *utils.lua*, *webs.lua* que permiten el uso extendido del canal de retorno de las aplicaciones que así lo necesiten.

3.5.1 DIAGRAMA DE CASOS DE USO

El Diagrama de Casos de uso general de la aplicación puede ser observado en la figura 3-16.

Figura 3-16: Diagrama de Casos de Uso general de la aplicación



Fuente: El Autor

3.5.2 MENÚ PRINCIPAL

La aplicación se carga en memoria en el Set Top Box y espera la acción del usuario para desplegar el menú de opciones de todas las aplicaciones desarrolladas, las que son fácilmente navegables y accedidas desde el control remoto, el video siempre esta mostrándose pudiendo re-dimensionarse para dar paso a más opciones de las aplicaciones interactivas.

3.5.2.1 Requisitos

3.5.2.1.1 Requisitos del Usuario

- El menú se despliega únicamente cuando el usuario lo decide, haciendo uso del mando a distancia. La aplicación inicia desplegando un mensaje en la parte superior derecha indicando la disponibilidad de contenido multimedia.
- Las opciones del menú deben ser claras y legibles para el usuario, cada

opción debe tener una descripción de su funcionalidad.

- Debe haber una opción que permita al usuario cerrar el menú para retornar al estado inicial de la aplicación para que en un momento deseado y mientras la aplicación se encuentre disponible el usuario puede acceder nuevamente.

3.5.2.1.2 Requisitos del sistema

- Algunas aplicaciones requieren canal de retorno, las aplicaciones que lo necesitan mostrarán al usuario cuando sea necesario conexión a Internet para el correcto funcionamiento de la aplicación.
- La aplicación iniciará a la orden del usuario mediante la acción de presionar el botón “ROJO” del control remoto.
- La última opción debe indicar la posibilidad de “Salir” accediendo a ella el menú se cerrará y la aplicación debe estar lista para ejecutarse de nuevo si se establece la acción del punto anterior.
- El menú se desplegará sobre el video de la programación de la televisora, la navegación sobre el menú de opciones se realizará con las teclas “FLECHA ARRIBA” y “FECHA ABAJO”, al estar posicionado en la última opción será posible ir de nuevo al principio y viceversa, la tecla “ENTER” accederá a la opción escogida desplegando la aplicación seleccionada, el video de la televisora se redimensiona al 40% de la pantalla y se traslada a la parte derecha de la misma, la aplicación en ejecución ocupara el 60% de la pantalla posicionándose al margen izquierdo.

3.5.2.2 Diagramas NCM

Cabe recalcar que los diagramas NCM son diagramas de estado que permiten describir de manera clara en que etapa se encuentra determinada acción de la aplicación.

El diagrama NCM para el menú ha sido dividido en varios diagramas a fin de hacer más fácil la comprensión y seguimiento de los mismos.

En la figura 3-17 se puede apreciar la leyenda utilizada para el diseño de los diagramas NCM de acuerdo con este estándar, resaltan los enlaces, puertos, nodos, contextos y las asignaciones.

Figura 3-17: Leyenda de los diagramas NCM

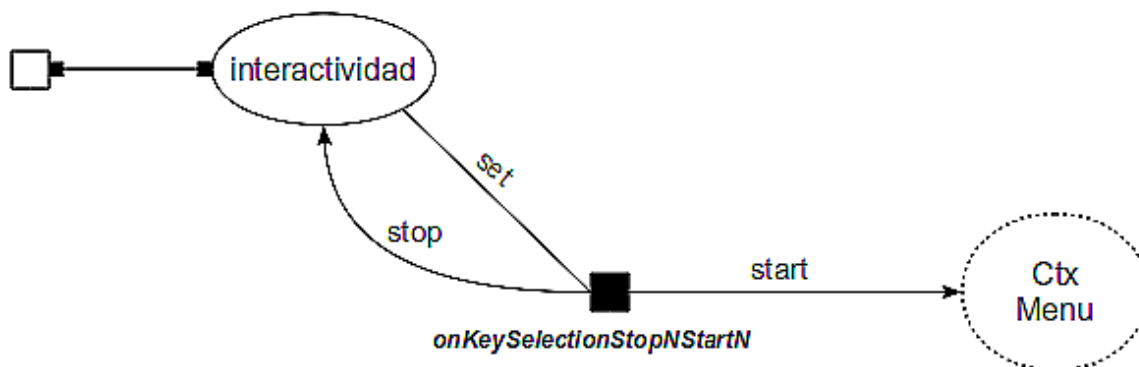


Fuente: El autor

En la figura 3-18 se gráfica la aplicación en su primer estado, cuando es lanzada queda activa a la espera de la interacción de los usuarios.

Una vez que se dispara el evento propiciado por el usuario, el contexto *Menu* es iniciado. Como el menú abarca las opciones que desplegarán las opciones y al ser su diagrama demasiado extenso para una mejor visión y comprensión, cada opción dentro del menú es representado en un diagrama NCM, estas imágenes son desde la figura 3-19 hasta la figura 3-24.

Figura 3-18: Diagrama NCM del momento inicial de la aplicación

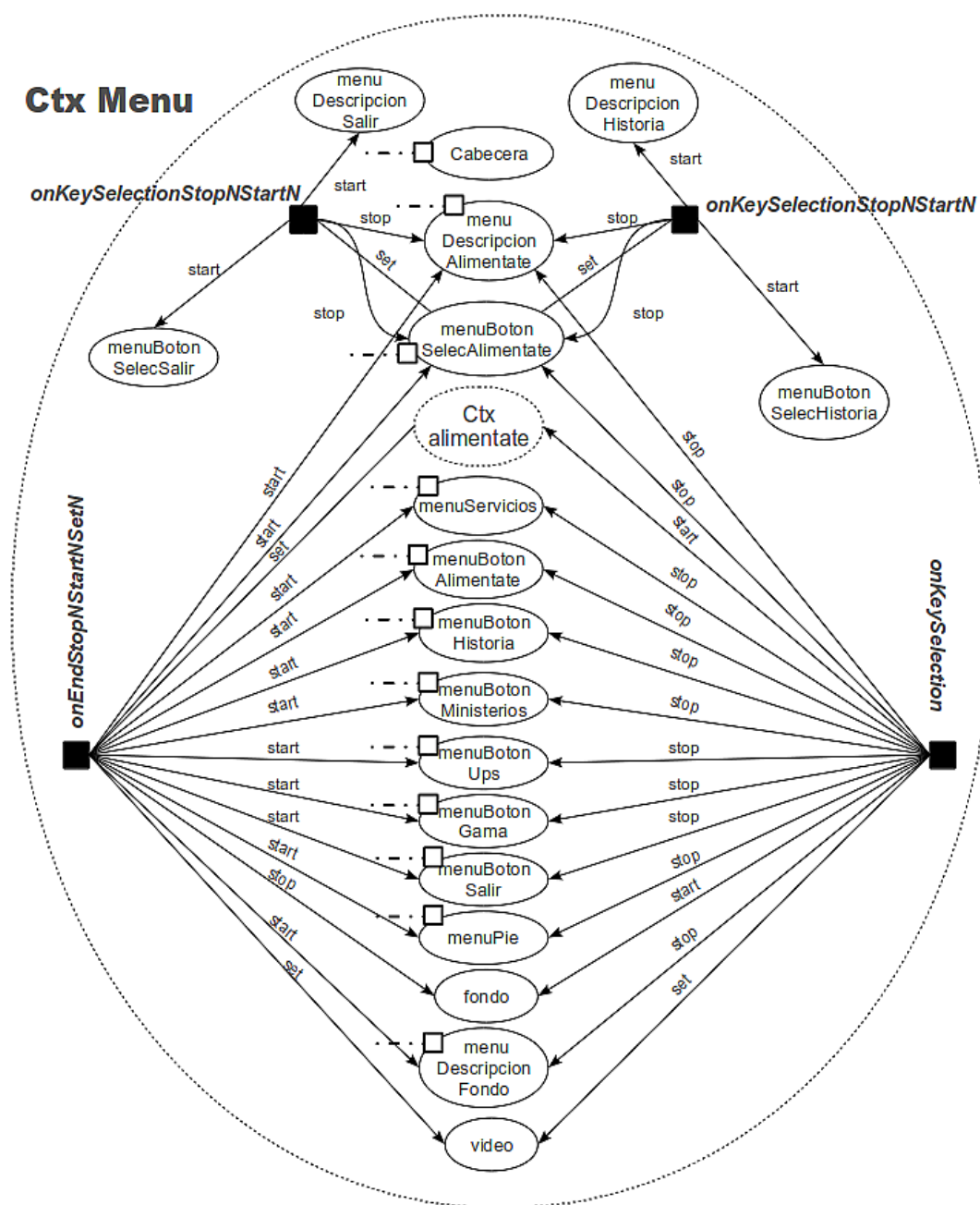


Fuente: El autor

En la figura 3-19 el Diagrama NCM dentro del contexto “Menu” cuando la opción Aliméntate Ecuador queda seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Salir”, desplazarse hacia abajo a la opción “Historia del Ecuador” o ingresar a esta opción, lo cual inicia el contexto “alimentate” que se amplía con más detalle en la sección 3.5.2.

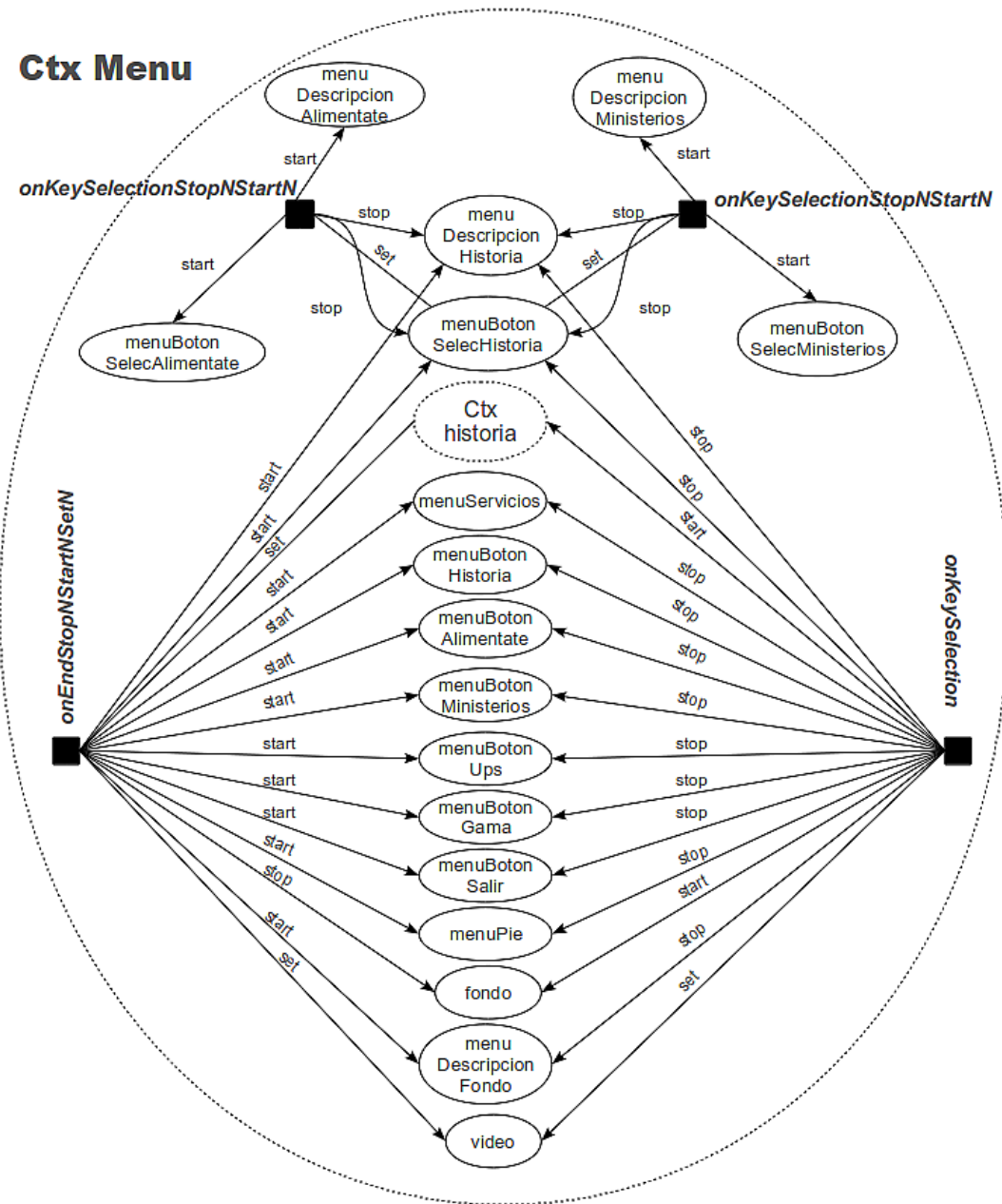
En la figura 3-20 el Diagrama NCM dentro del contexto “Menu” cuando la opción “Historia del Ecuador” queda seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Aliméntate Ecuador”, desplazarse hacia abajo a la opción “Ministerios” o ingresar a esta opción, lo cual inicia el contexto “historia” que se amplía con más detalle en la sección 3.5.3.

Figura 3-19: Diagrama NCM, Contexto Menú, opción Aliméntate Ecuador



Fuente: El autor

Figura 3-20: Diagrama NCM, Contexto Menú, opción Historia del Ecuador

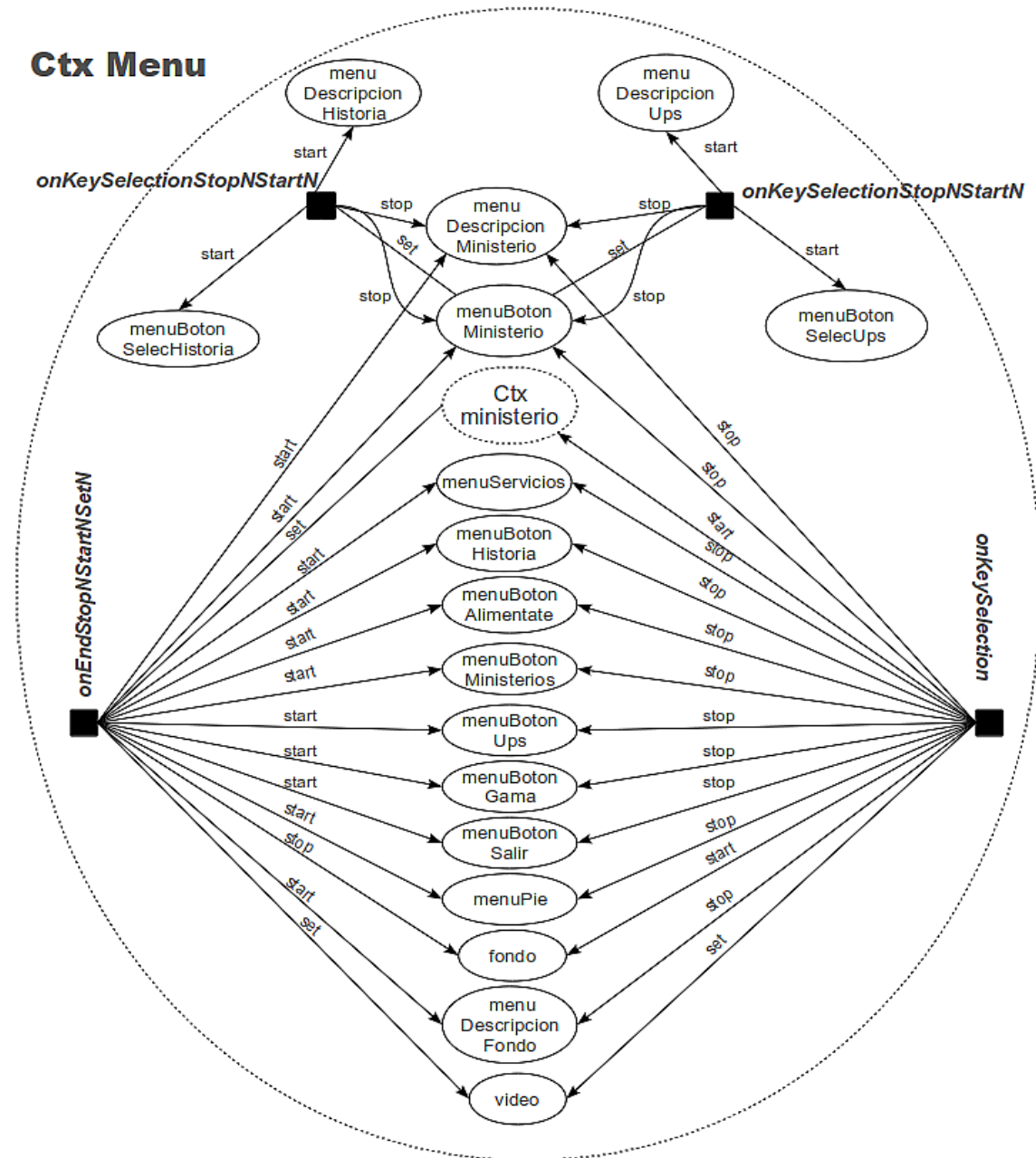


Fuente: El autor

En la figura 3-21 el Diagrama NCM dentro del contexto “Menu” cuando la opción “Entidades Públicas” queda seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Historia del Ecuador”, desplazarse hacia abajo a la opción “Ups” o ingresar a esta opción, lo cual inicia el contexto “ministerios” que se amplía con más detalle en la

sección 3.5.4.

Figura 3-21: Diagrama NCM, Contexto Menú, opción Entidades Públicas

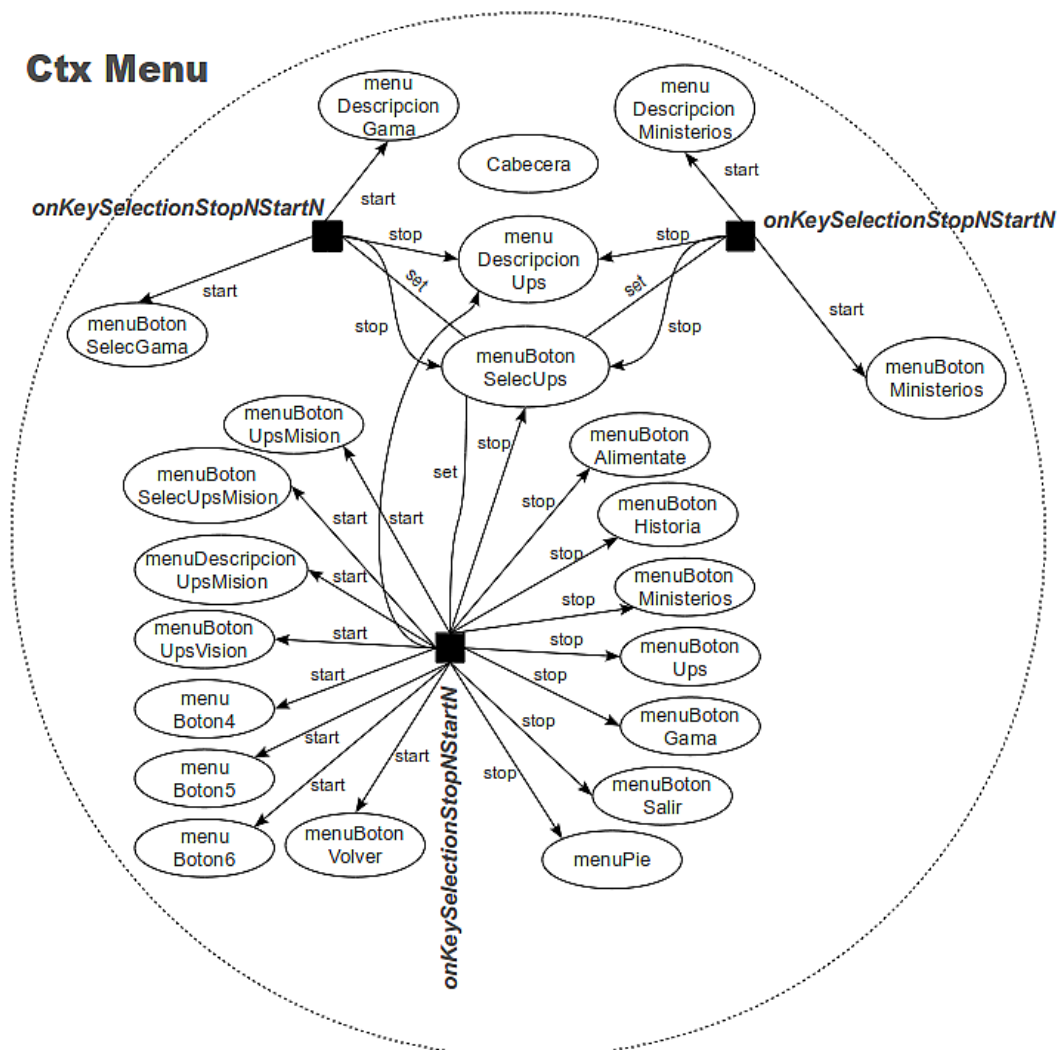


Fuente: El autor

En la figura 3-22 el Diagrama NCM dentro del contexto “Menu” cuando la opción “Ups” queda seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Ministerios”,

desplazarse hacia abajo a la opción “GamaTv” o ingresar a esta opción, lo cual lleva a un segundo menú con las opciones “Misión”, “Visión” y “Volver”.

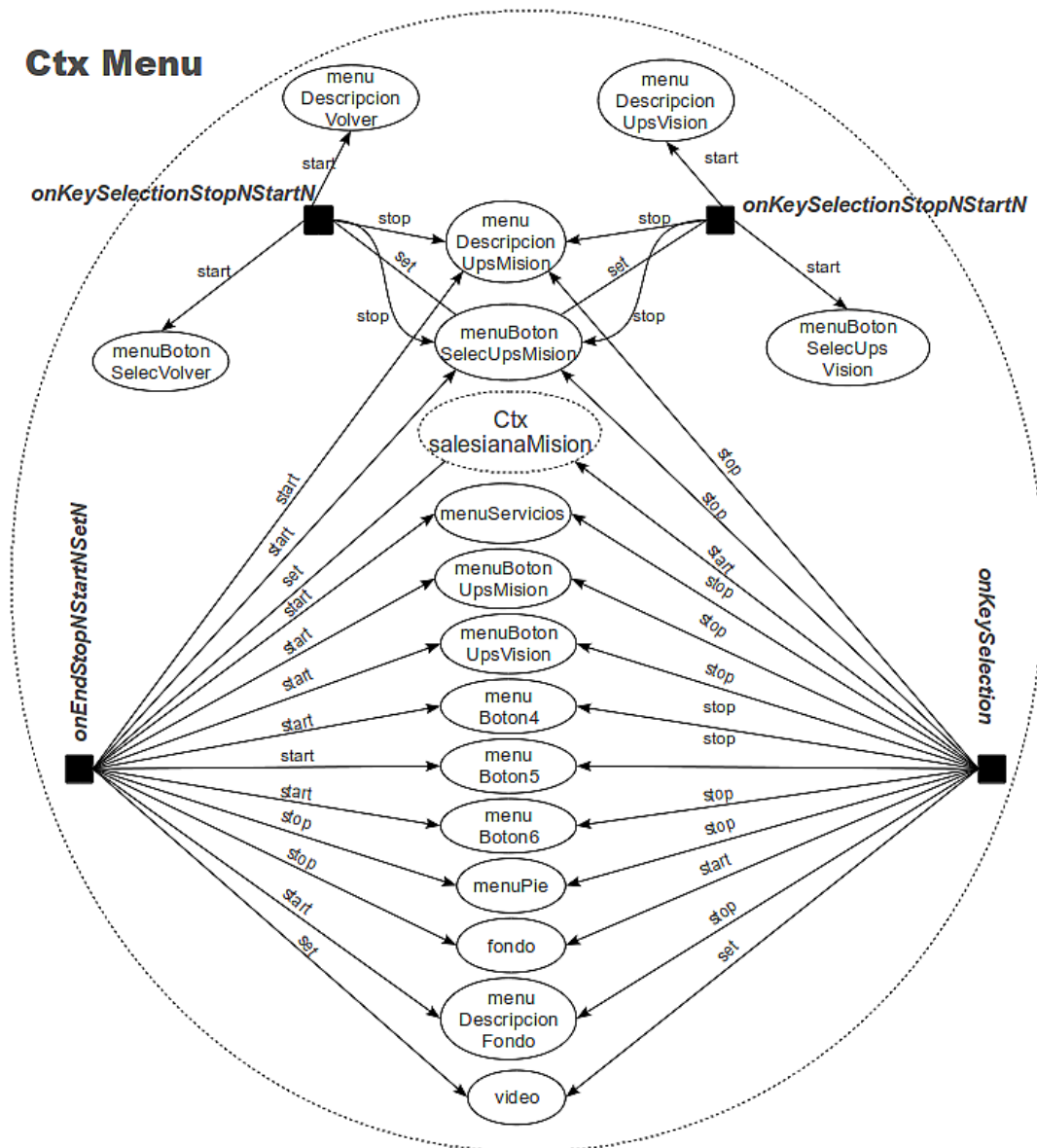
Figura 3-22: Diagrama NCM, Contexto Menú, opción Ups



Fuente: El autor

En la figura 3-23 el Diagrama NCM dentro del contexto “Menu”, dentro de la opción “Ups” la opción “Misión” queda seleccionada, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Regresar”, desplazarse hacia abajo a la opción “Visión” o ingresar a esta opción, lo cual inicia el contexto “salesianaMision” que se amplía con más detalle en la sección 3.5.6.

Figura 3-23: Diagrama NCM, contexto Menu, opción Misión



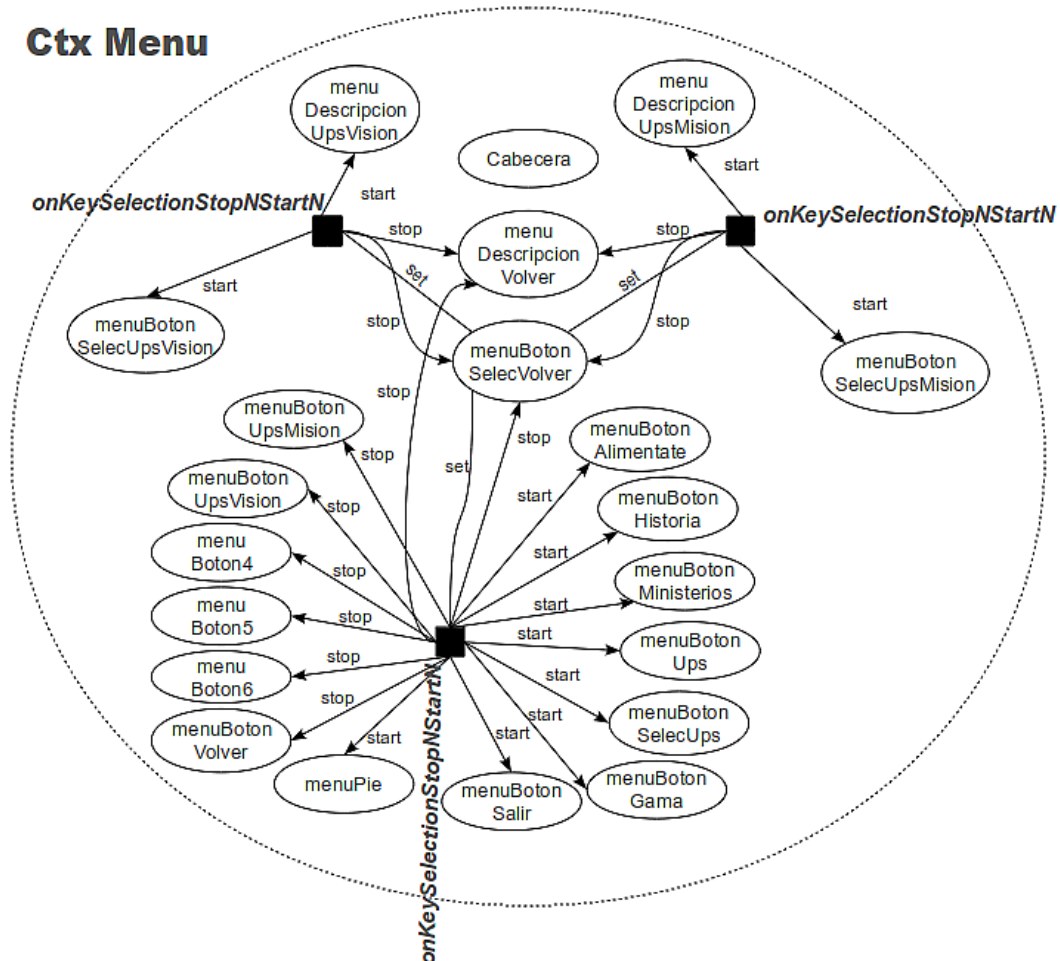
Fuente: El autor

En la figura 3-24 el Diagrama NCM dentro del contexto “Menu”, dentro de la opción “Ups” cuando la opción “Visión” es seleccionada, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Misión”, desplazarse hacia abajo a la opción “Regresar” o ingresar a esta opción, lo cual inicia el contexto “salesianaVision”.



En la figura 3-25 el Diagrama NCM dentro del contexto “*Menu*”, dentro de la opción “*Ups*” cuando la opción “*Regresar*” es seleccionada, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “*Visión*”, desplazarse hacia abajo a la opción “*Misión*” o ingresar a esta opción para regresar al primer menú con la opción “*Ups*” marcada.

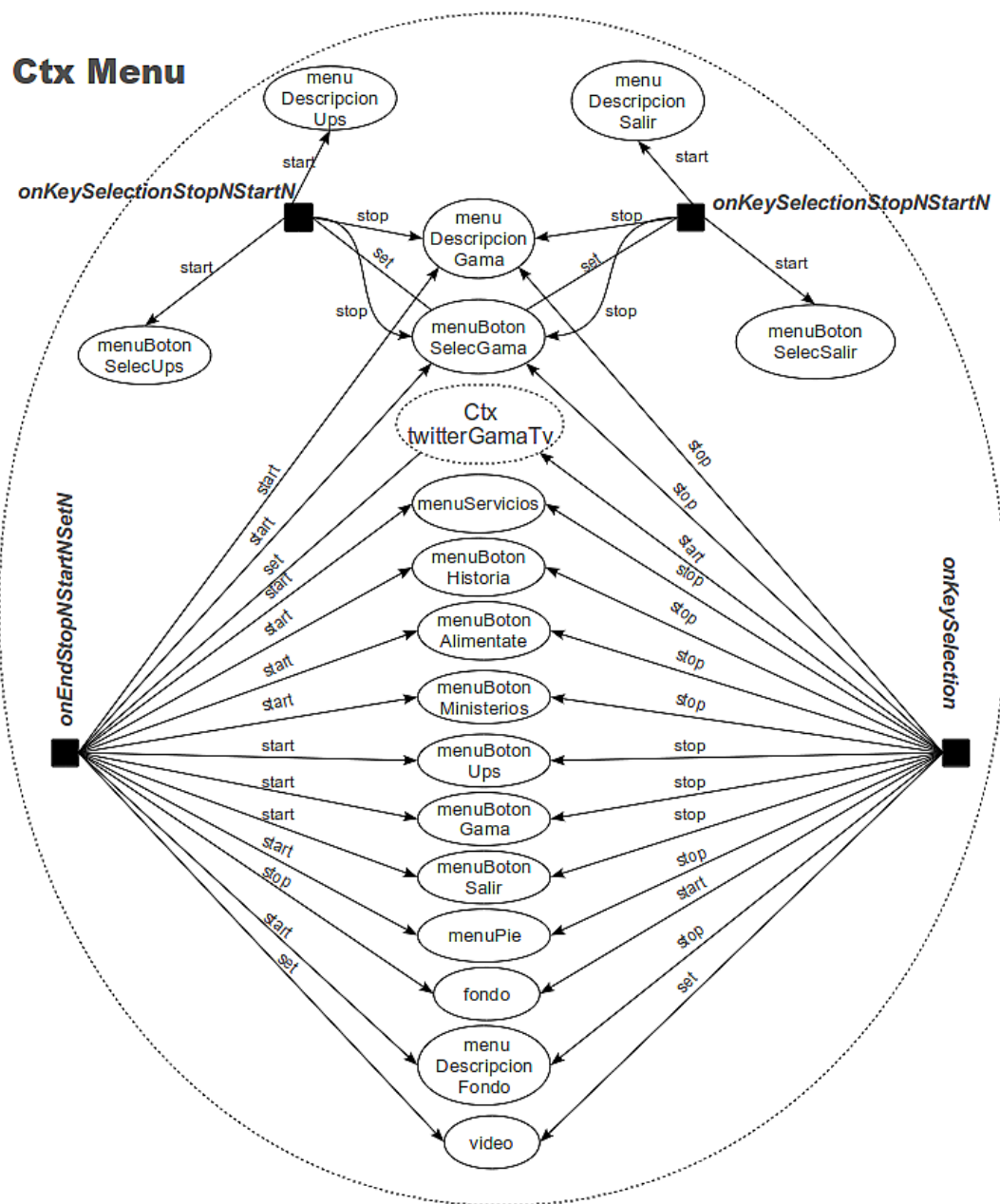
Figura 3-25: Diagrama NCM, contexto Menu, opción Regresar



Fuente: El autor

En la figura 3-26 el Diagrama NCM dentro del contexto “Menu” cuando la opción “GamaTv” es seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “Ups”, desplazarse hacia abajo a la opción “Salir” o ingresar a esta opción, lo cual inicia el contexto “twitterGamaTv” que se amplía con más detalle en la sección 3.5.6.

Figura 3-26: Diagrama NCM, Contexto Menú, opción GamaTv

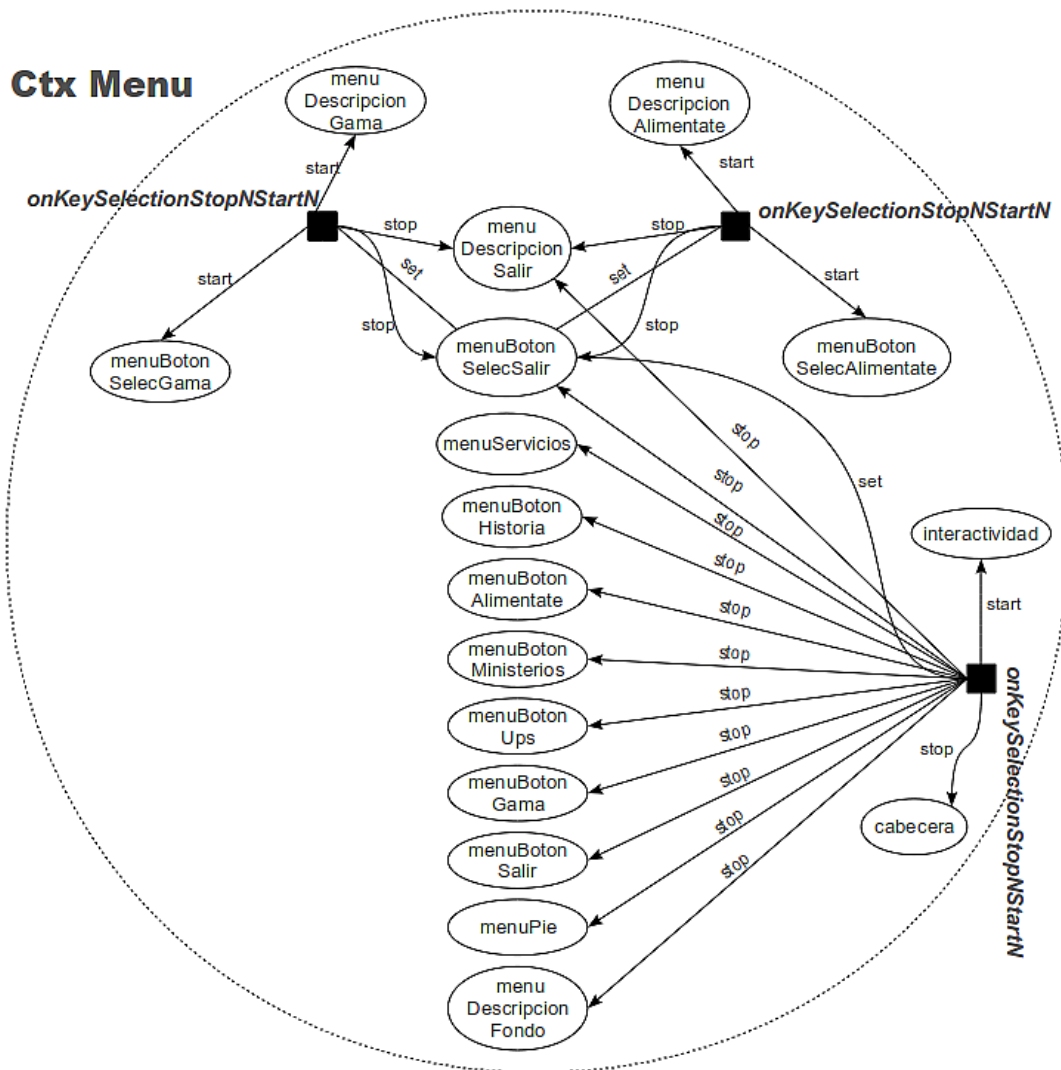


Fuente: El autor

En la figura 3-27 el Diagrama NCM dentro del contexto “Menu” cuando la opción “Salir” es seleccionada al mostrarse todas las opciones, su descripción también es activada. En ese estado se tiene tres opciones correspondientes a la interacción del usuario, desplazarse hacia arriba a la opción “GamaTv”, desplazarse hacia abajo a la opción “Alimentate Ecuador” o ingresar a esta opción, lo cual cierra el

menú principal y la aplicación queda en estado inicial a la espera nuevamente de la interacción del usuario.

Figura 3-27: Diagrama NCM, Contexto Menú, opción Salir

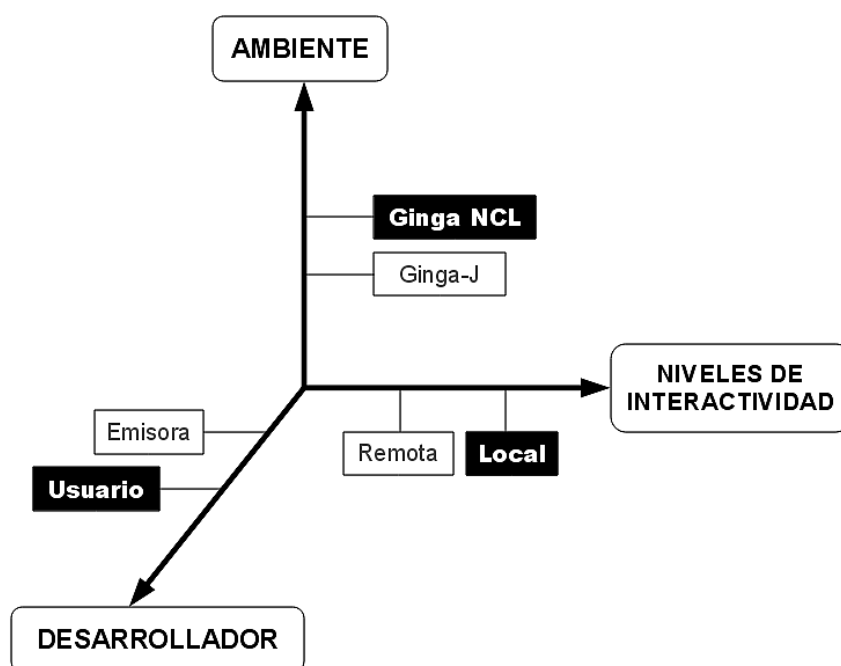


Fuente: El autor

3.5.2.3 Diagramas de desarrollo

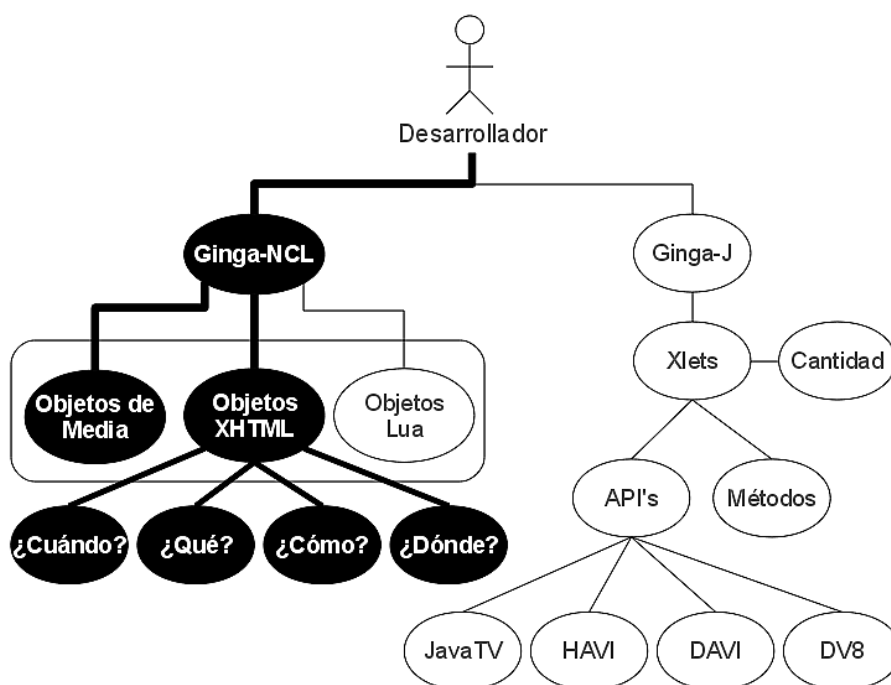
En la figura 3-28 se puede apreciar el nivel de interactividad, el contexto del desarrollador y los ambientes utilizados. En la figura 3-29 refleja las opciones de desarrollo para este punto de la aplicación.

Figura 3-28: Localización espacial de la opción del menú de la aplicación



Fuente: El autor

Figura 3-29: Diagrama de opciones de desarrollo, opción menú



Fuente: El autor

3.5.2.4 Implementación

El menú esta programado exclusivamente en NCL, tiene 776 líneas de código llamado *main.ncl*.

3.5.3 APLICACIÓN 1. ALIMÉNTATE ECUADOR

La aplicación despliega tres opciones de platos de comida con fotografías ilustrativas, mediante el mando a distancia escoge uno de los platos, el programa en seguida le muestra la elección que ha hecho y le enseña cuan buena o mala fue su decisión.

3.5.3.1 Requisitos

3.5.3.1.1 Requisitos del Usuario

La opción Aliméntate Ecuador es la primera en el menú de opciones, de tal manera que es la primera en seleccionarse.

Las fotografías de los platos deben ser claras y legibles para el usuario, cada opción debe tener una descripción de su contenido.

Cuando el usuario escoja su opción, el resultado de su elección deberá aparecer inmediatamente, ocultando las demás opciones, deberá haber una opción para retornar a la primera vista para repetir la rutina.

Todas las indicaciones de como manipular la aplicación deberán aparecer en la zona de “*descripción*”, de igual forma debe ser clara y legible para el usuario.

Debe haber una opción para cerrar la aplicación y regresar al menú principal.

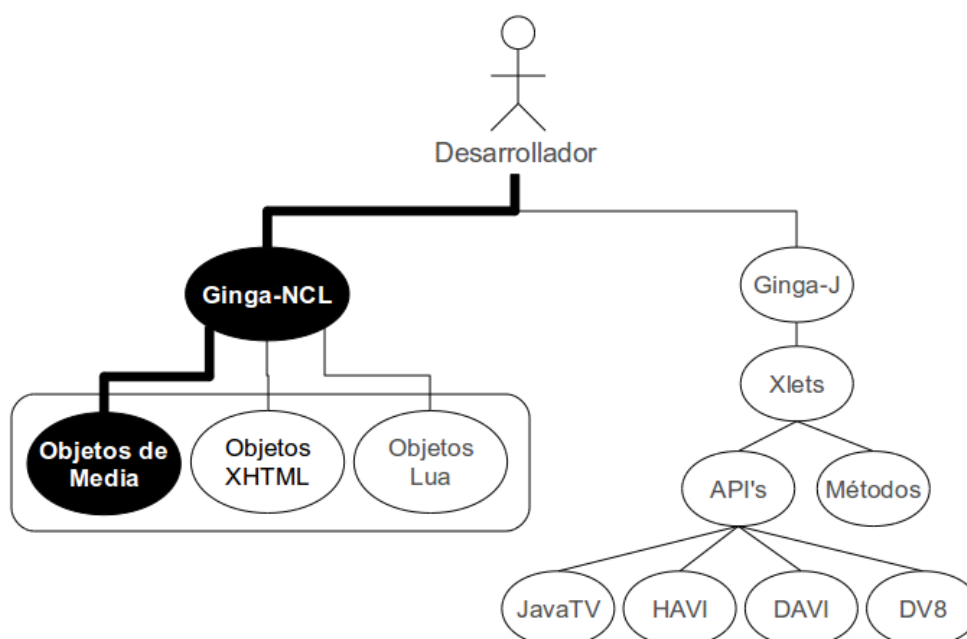
3.5.3.1.2 Requisitos del sistema

La aplicación iniciará cuando el usuario escoja la aplicación Aliméntate Ecuador seleccionándola del menú principal si aún no lo está y presionando la tecla

"ENTER" del control remoto y la aplicación se ejecutará.

Con los botones "AMARILLO", "AZUL" "VERDE" se escoge una opción de platillo, la aplicación ocultará las opciones que no fueron tomadas en cuenta y dejará solo la opción elegida y mostrará el resultado de su elección, con la tecla "CURSOR LEFT" se regresará al estado inicial, ocultando el resultado de la elección y mostrando todos los platos para empezar nuevamente.

Figura 3-30: Diagrama de opciones de desarrollo, opción Aliméntate Ecuador



Fuente: El autor

3.5.3.2 Diagramas NCM

El diagrama NCM se muestra en la figura 3-31.

3.5.3.3 Diagramas de desarrollo

El nivel de interactividad, el contexto del desarrollador y los ambientes utilizados son los mismos que los mostrados en la figura 3-28 de la sección del menú. En la figura 3-30 refleja las opciones de desarrollo.

3.5.3.4 Implementación

La aplicación Aliméntate Ecuador está programada en NCL, se encuentra en un archivo separado del menú que se llama *alimentate.ncl* tiene alrededor de 203 líneas de código.

3.5.4 APLICACIÓN 2. HISTORIA DEL ECUADOR

La aplicación presenta una serie de preguntas con diferentes opciones de respuestas acerca de la historia más importante de nuestro país, en cualquier momento el usuario puede revisar en cual de las preguntas acertó y cuales se equivocó.

3.5.4.1 Requisitos

3.5.4.1.1 Requisitos del usuario

La aplicación Historia del Ecuador es la segunda opción en el menú principal, el usuario accede a ella desplazándose por el menú.

Tanto como las preguntas y las respuestas deben ser claramente legibles para el usuario, las preguntas o respuestas que sobrepasen la longitud del espacio de la aplicación deberán mostrarse en n líneas de forma alineada.

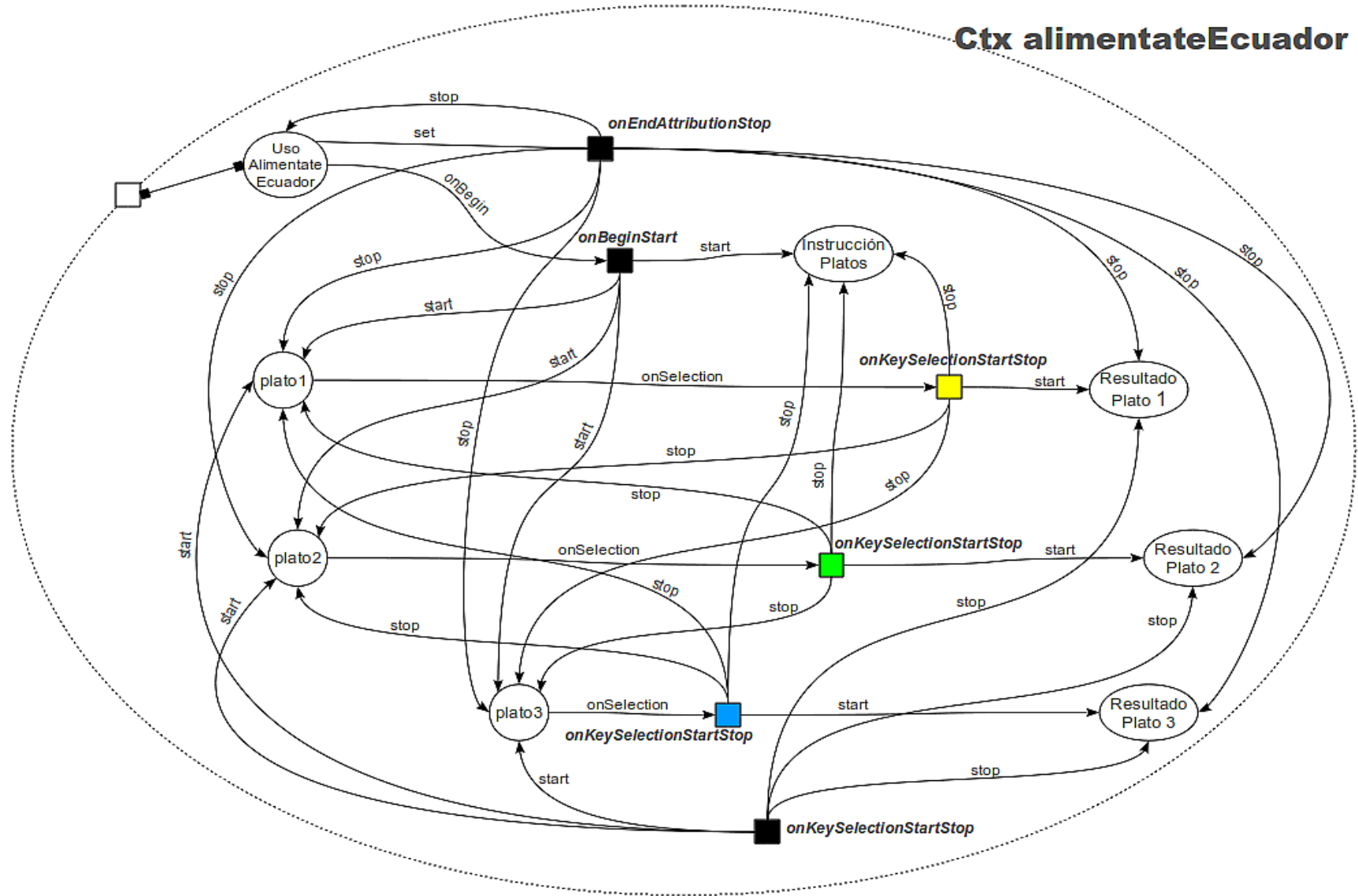
Las respuestas se escogen mediante los botones del control remoto así como también la visualización de las respuestas, también debe haber una opción para cerrar la aplicación y retornar al menú principal.

Las preguntas y respuestas deben estar en un archivo plano separado a fin de ir ingresando, actualizando o borrando varias de las mismas.

Todas las indicaciones de como manipular la aplicación deberán aparecer en la zona de “*descripción*”, de igual forma debe ser clara y legible para el usuario.

Debe haber una opción para cerrar la aplicación y regresar al menú principal.

Figura 3-31: Diagrama NCM, contexto Aliméntate Ecuador



Fuente: El autor

3.5.4.1.2 Requisitos del sistema

La aplicación iniciará cuando el usuario escoja la opción Historia del Ecuador seleccionándola del menú principal si aún no lo está y presionando la tecla "ENTER" del control remoto la aplicación se ejecutará.

La primera pregunta aparecerá y las posibles respuestas, para escoger una de ellas se puede usar los botones "1", "2", "3" o "4" del control remoto. Para avanzar a la siguiente pregunta se usará el botón "CURSOR RIGTH", así mismo para retroceder entre las preguntas se usará el botón "CURSOR LEFT".

Para observar el resultado de aciertos o errores se usará el botón "ENTER", y para finalizar la aplicación se usará el botón "ROJO" que regresará al menú principal. Las preguntas y respuestas son leídas de un archivo separado ubicado en la misma ruta que el de la aplicación.

3.5.4.2 Diagramas NCM

El diagrama NCM se muestra en la figura 3-32.

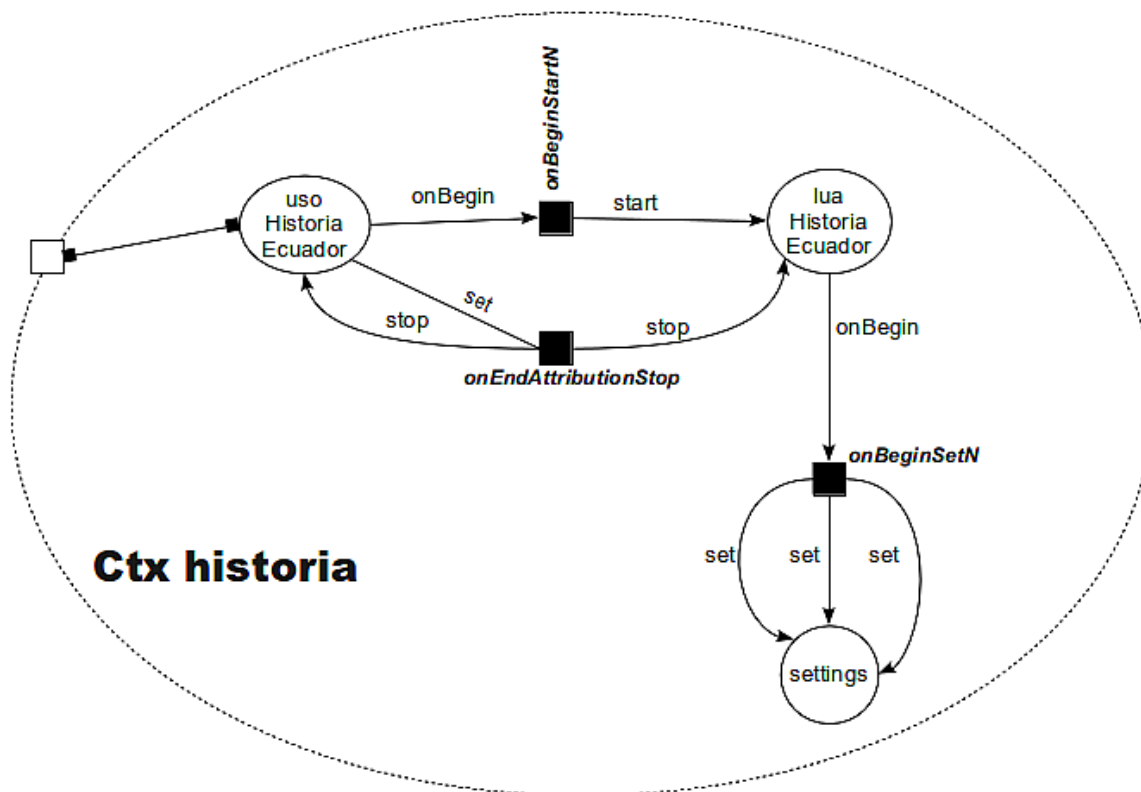
3.5.4.3 Diagramas de desarrollo

El nivel de interactividad, el contexto del desarrollador y los ambientes utilizados son los mismos que los mostrados en la figura 3-28 de la sección del menú. En la figura 3-33 refleja las opciones de desarrollo.

3.5.4.4 Implementación

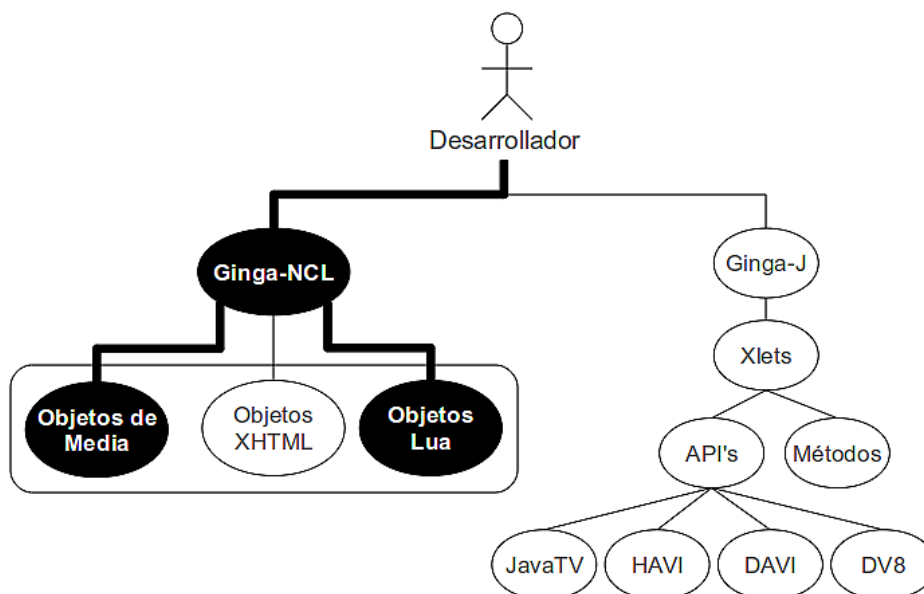
La aplicación Historia del Ecuador está programada en NCL y Lua, se encuentra en un archivo separado del menú que se llama *historia.ncl* tiene alrededor de 84 líneas de código. En esta aplicación el archivo NCL únicamente sirve para definir el contexto que se desata cuando es lanzado desde el evento que produce su selección en el menú, de la misma manera para cuando se cierra la aplicación y hay que notificarlo al contexto para regresar al menú. El corazón de la aplicación radica en un archivo llamado *historiaEcuador.lua* donde ocurre toda la acción. Este archivo tiene alrededor de 263 líneas de código. A estos se suman dos archivos más, donde se encuentran las preguntas y respuestas llamado *preguntas.lua* y otro llamado *config.lua* de donde se formatean las variables de la información que se encuentra en archivo de preguntas citado anteriormente.

Figura 3-32: Diagrama NCM de la aplicación Historia del Ecuador



Fuente: El autor

Figura 3-33: Diagrama de opciones de desarrollo, Historia del Ecuador



Fuente: El autor

3.5.5 APLICACIÓN 3. MINISTERIOS

La aplicación se conecta con un *Web Service*⁵⁵ del Sistema de Actas de Finiquito del Ministerio de Relaciones Laborales para consultar información sobre un turno en este sistema.

3.5.5.1 Requisitos

3.5.5.1.1 Requisitos del usuario

La aplicación Entidad es la tercera opción en el menú principal, el usuario accede a ella desplazándose por el menú.

La aplicación permite al usuario ingresar su número de cédula y buscar si existe un turno en el Sistema de Actas de Finiquito, la información que se puede desplegar puede ser la siguiente:

Que el usuario no exista o nunca haya sido ingresado al Sistema de Actas de

⁵⁵ Un Web Service es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Finiquito. En este caso un mensaje indicando este particular aparecerá en pantalla.

Que el usuario tenga turnos anteriores a la fecha de búsqueda registrado en el Sistema de Actas de Finiquito. En este caso un mensaje indicando este particular aparecerá en pantalla.

Que el usuario tenga un turno vigente registrado en el Sistema de Actas de Finiquito, en este caso los datos que se mostrarán son los siguientes: Ciudad, Inspector y Fecha y hora del turno.

3.5.5.1.2 Requisitos del sistema

La aplicación iniciará cuando el usuario escoja la opción Entidades Públicas seleccionándola del menú principal si aún no lo está y presionando la tecla "ENTER" del control remoto, la aplicación se ejecutará.

La aplicación requiere conexión a Internet, en este caso una salida ethernet para el Set Top Box, la aplicación en el caso de que no pueda conectarse al *web service* por falla en la conectividad o porque el receptor no cuenta con la interfaz para el canal de retorno mostrará un mensaje de error.

La aplicación permite el ingreso de un número de cédula, para lo cual deberá usar los botones numéricos del control remoto, esto es, los botones 1, 2, 3, 4, 5, 6, 7, 8, 9, 0; con el botón "CURSOR LEFT" puede ir borrando los números desde el último al primero en el orden de ingreso para corregir si uno de ellos no es el indicado; el botón "ENTER" es el que permite que la búsqueda se realice, la conexión con el *web service* se llevará a cabo.

En el ingreso del número de cédula de ciudadanía ocurren las siguientes validaciones:

Para el ingreso solo se permiten números, estos botones corresponden al teclado numérico, como máximo diez dígitos son permitidos.

Si no se ha ingresado el número de dígitos establecido, el botón “ENTER” no responderá a ninguna acción.

Cuando la cantidad de dígitos sea la correcta, se visualizará en pantalla un mensaje que advertirá al usuario que puede continuar con la búsqueda, el botón “ENTER” queda a la espera de la interacción del usuario.

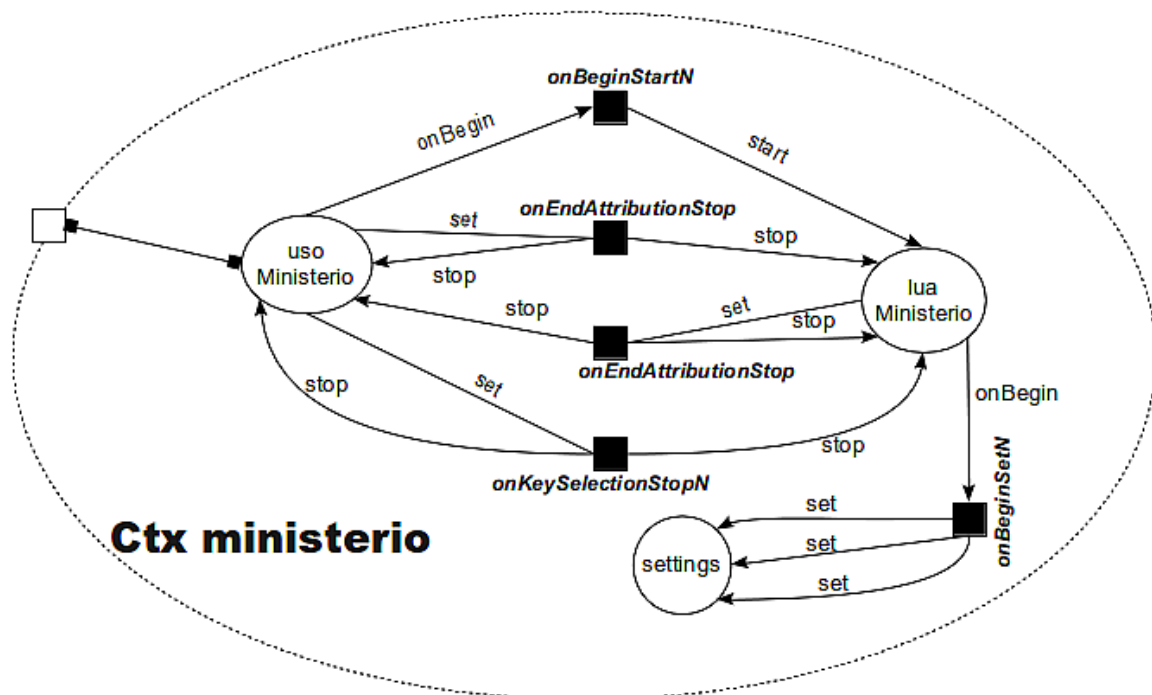
3.5.5.2 Diagramas NCM

El diagrama NCM se muestra en la figura 3-34.

3.5.5.3 Diagramas de desarrollo

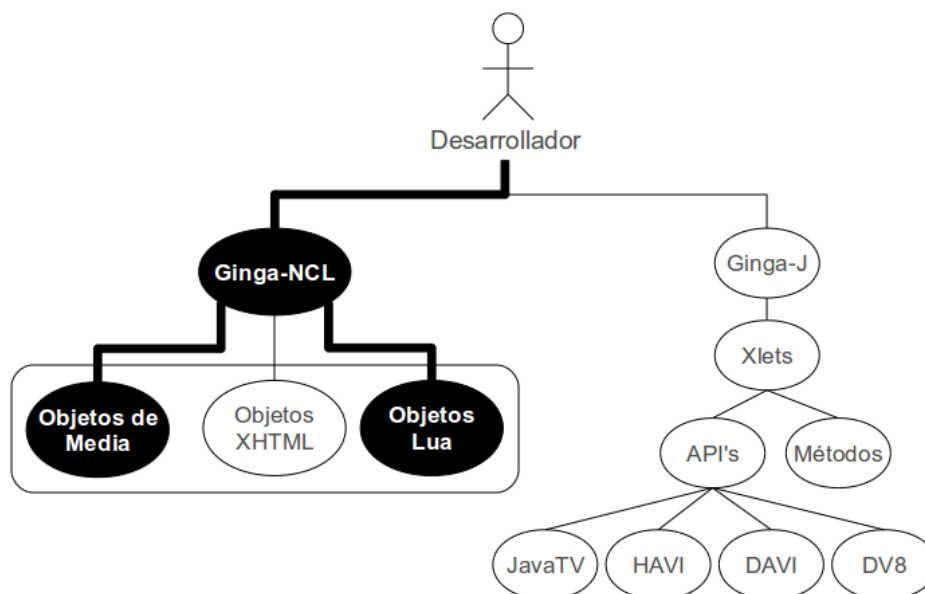
El nivel de interactividad, el contexto del desarrollador y los ambientes utilizados son los mismos que los mostrados en la figura 3-28 de la sección del menú. En la figura 3-35 refleja las opciones de desarrollo.

Figura 3-34: Diagrama NCM de la aplicación Entidades Públicas



Fuente: El autor

Figura 3-35: Diagrama de opciones de desarrollo, Entidades Públicas



Fuente: El autor

3.5.5.4 Implementación

La aplicación Entidades Públicas está programada en NCL y Lua, se encuentra en un archivo separado del menú que se llama *ministerio.ncl* tiene alrededor de 95 líneas de código. El archivo NCL únicamente sirve para definir el contexto que se desata cuando es lanzado desde el evento que produce su selección en el menú, de la misma manera para cuando se cierra la aplicación y hay que notificarlo al contexto para regresar al menú. El archivo *ministerio.lua* se encarga de recoger los datos que el usuario ingrese, controlar las condiciones de este ingreso, permitir la ejecución del *Web Service* y muestra en pantalla los resultados de esta búsqueda. Este archivo tiene alrededor de 280 líneas de código.

El archivo *ws.lua* se ejecuta cuando se recibe un evento de teclado del botón “ENTER” para comenzar la conexión con el *Web Service*, en este archivo está la configuración de conexión y el procesamiento posterior a la respuesta que retorne el *Web Service*.

En la norma brasileña especificada para el manejo de los protocolos de conexión

de Internet, solo el protocolo TCP está disponible como módulo para las aplicaciones desarrolladas con Lua, es por esta razón que para el diseño de esta aplicación se hizo uso de un módulo desarrollado por Manoel Campos da Silva, programador brasileño llamado *NCLua SOAP v0.5.6.6*⁵⁶ disponible con licencia Creative Commons⁵⁷ que facilita en gran medida el procesamiento de peticiones Soap que es el protocolo para el manejo de WebServices.

3.5.6 APLICACIÓN 4. GAMATV.

La aplicación accede a la cuenta del twitter de GamaTv y muestra los últimos mensajes, la aplicación requiere canal de retorno, en este caso salida a Internet.

3.5.6.1 Requisitos

3.5.6.1.1 Requisitos del usuario

La aplicación GamaTv es la cuarta opción en el menú principal, el usuario accede a ella desplazándose por el menú.

La aplicación debe mostrar el estado de la conexión y su visualización debe ser clara y legible así como también el texto de los twitts.

Todas las indicaciones de como manipular la aplicación deberán aparecer en la zona de “*descripción*”, de igual forma debe ser clara y legible para el usuario.

Debe haber una opción para cerrar la aplicación y regresar al menú principal.

3.5.6.1.2 Requisitos del sistema

La aplicación iniciará cuando el usuario escoja la opción GamaTv seleccionándola del menú principal si aún no lo está y presionando la tecla "ENTER" del control remoto, la aplicación se ejecutará.

⁵⁶ Módulo diseñado exclusivamente en Lua para dar soporte al protocolo Soap en aplicaciones de Televisión Digital.

⁵⁷ Licencia de software libre para documentos y desarrollos

La aplicación requiere conexión a Internet, en este caso una salida *ethernet* para el Set Top Box, la aplicación en el caso de que no pueda conectarse a la cuenta de twitter de GamaTv por falla en la conectividad o porque el receptor no cuenta con la interfaz para el canal de retorno mostrará un mensaje de error al conectarse.

Si la conexión se realizó con éxito, aparecerán los dos últimos mensajes cortos de twitter, si se quiere visualizar mensajes anteriores a los que se están mostrando se puede usar el botón “CURSOR LEFT” para desplegarlos en pantalla.

Para finalizar la aplicación el botón “ROJO” permitirá regresará al menú principal.

3.5.6.2 Diagramas NCM

El diagrama NCM se muestra en la figura 3-36.

3.5.6.3 Diagramas de desarrollo

El nivel de interactividad, el contexto del desarrollador y los ambientes utilizados son los mismos que los mostrados en la figura 3-28 de la sección del menú. En la figura 3-33 refleja las opciones de desarrollo.

3.5.6.4 Implementación

La aplicación GamaTv está programada en NCL y Lua, se encuentra en un archivo separado del menú que se llama *gamaTv.ncl* tiene alrededor de 47 líneas de código. El archivo NCL únicamente sirve para definir el contexto que se desata cuando es lanzado desde el evento que produce su selección en el menú, de la misma manera para cuando se cierra la aplicación y hay que notificarlo al contexto para regresar al menú. El corazón de la aplicación radica en un archivo llamado *gamatv.lua* donde ocurre toda la acción. Este archivo tiene alrededor de 178 líneas de código. Además existen tres archivos de suma importancia que son: *tcp.lua*, *LectorRSS.lua* y *Webs.lua*. El primer archivo *tcp.lua* esta destinado a la configuración de la conexión, desconexión y eventos de lectura de datos. El

segundo archivo *LectorRSS.lua* permite que la data recibida sea formateada para la correcta presentación en pantalla. El tercer archivo *Webs.lua* sirve para las distintas opciones de conexión y es donde se establecen las variables de las cuentas de twitter.

3.5.7 APLICACIÓN 5. UPS

La aplicación muestra información relacionada con la Universidad Politécnica Salesiana, su misión y visión

3.5.7.1 Requisitos

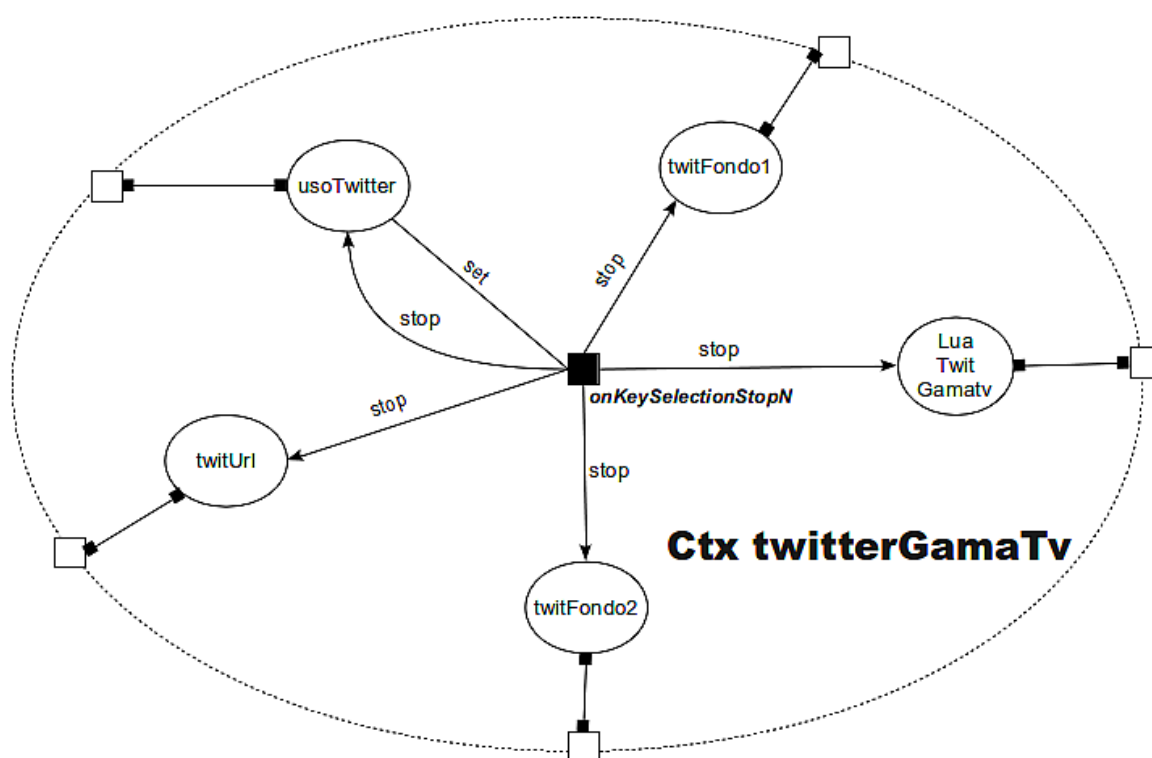
3.5.7.1.1 Requisitos del usuario

La aplicación UPS es la quinta opción en el menú principal, el usuario accede a ella desplazándose por el menú.

La aplicación debe mostrar un segundo menú con las opciones, Misión, Visión y regresar, la misión y la visión debe mostrarse de manera adecuada y legible. Todas las indicaciones de como manipular la aplicación deberán aparecer en la zona de “*descripción*”, de igual forma debe ser clara y legible para el usuario.

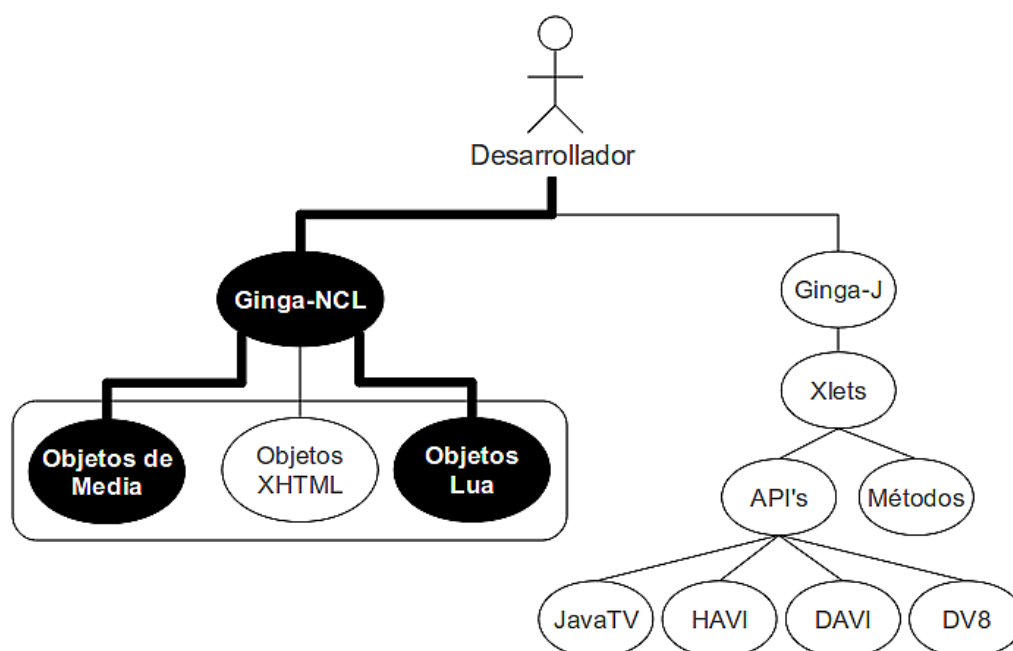
Debe haber una opción para cerrar la aplicación y regresar al menú principal.

Figura 3-36: Diagrama NCM de la aplicación GamaTv



Fuente: El autor

Figura 3-37: Diagrama de opciones de desarrollo, opción GamaTv



Fuente: El autor

3.5.7.1.2 Requisitos del sistema

La aplicación iniciará cuando el usuario escoja la opción UPS seleccionándola del menú principal si aún no lo está y presionando la tecla "ENTER" del control remoto, la aplicación se ejecutará.

La aplicación mostrará un segundo menú que podrán ser recorridos con los botones "CURSOR UP" y "CURSOR DOWN", con el botón "ENTER" en las dos primeras acciones desplegará en pantalla la misión o visión según sea el caso, al escoger una de estas opciones, el video se redimensiona y se alinea a la izquierda. Para retornar el botón "ROJO" permite mostrar nuevamente el video en pantalla completa y visualizar el segundo menú.

La tercera opción(Regresar) permite a la aplicación finalizar este contexto y retornar al menú principal.

3.5.7.2 Diagramas NCM

El diagrama NCM se muestra en la figura 3-39.

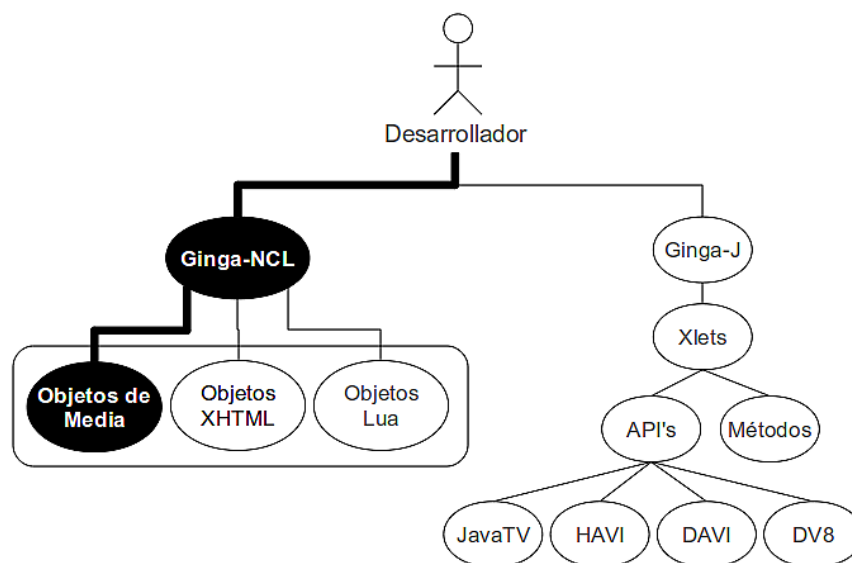
3.5.7.3 Diagramas de desarrollo

El nivel de interactividad, el contexto del desarrollador y los ambientes utilizados son los mismos que los mostrados en la figura 3-28 de la sección del menú. En la figura 3-38 refleja las opciones de desarrollo.

3.5.7.4 Implementación

La aplicación UPS está programada en NCL, se encuentra en un archivo separado del menú que se llama *ups.ncl*, tiene alrededor de 85 líneas de código. En este caso se enfatiza los contextos, ya que se tiene creados uno para cada caso, el contexto de la Misión y el contexto de la visión.

Figura 3-38: Diagrama de opciones de desarrollo, opción Ups



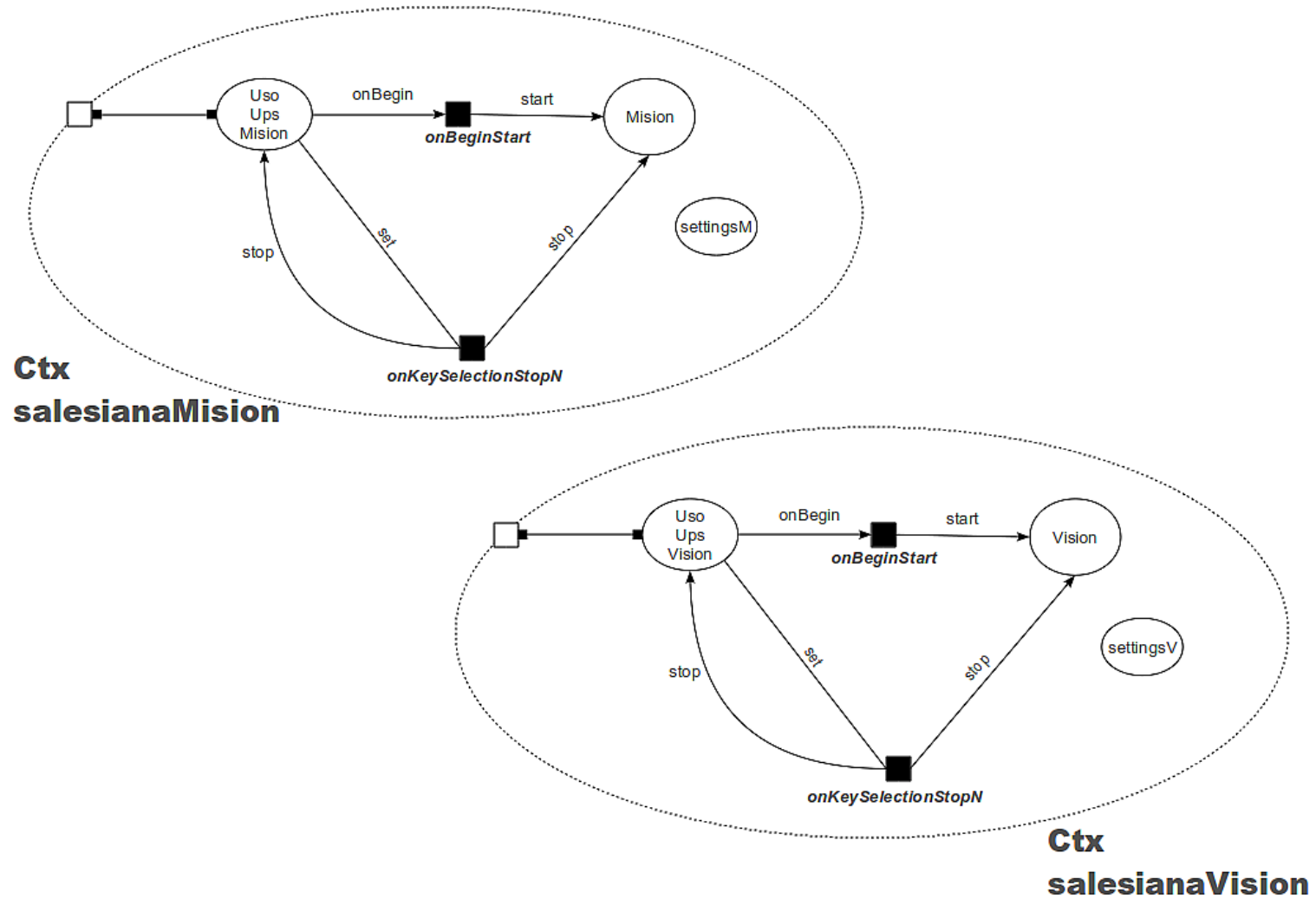
Fuente: El autor

3.6 CANAL DE RETORNO

El canal de retorno es el canal de comunicación entre un set top box y la nube, es decir, la transmisión de datos se produce en ambas direcciones.

Estos canales de retorno ya están presentes en set-top boxes a través de la interfaz Ethernet, como se ve en el modelo de equipo de Proview XPS-1000 (Proview, 2008) y Zinwell Modelo ZDS-620S (Zinwell, 2008). En la Figura 3-40 se puede apreciar el canal de comunicación de la estación televisiva con la nube (véase B) y otros puntos posibles de comunicación (punto C). Es de suma importancia tener en cuenta todos los posibles puntos de comunicación de los receptores, porque, cuando la aplicación se esté ejecutando en el receptor sus datos pueden viajar a la nube, es decir, el canal de comunicación puede conectarse a un servicio en un centro de datos (base de datos, por ejemplo, servicios web, etc.). Un diagrama más detallado se muestra en la Figura 3-41.

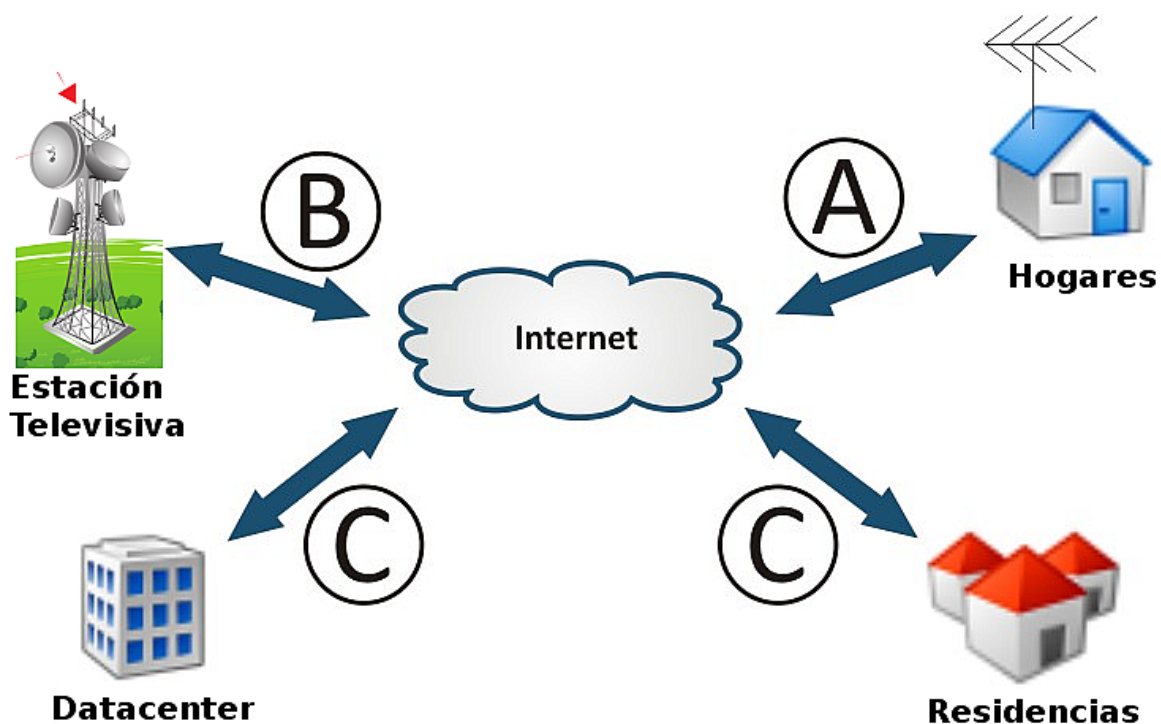
Figura 3-39: Diagrama NCM de la aplicación Ups



Fuente: El autor

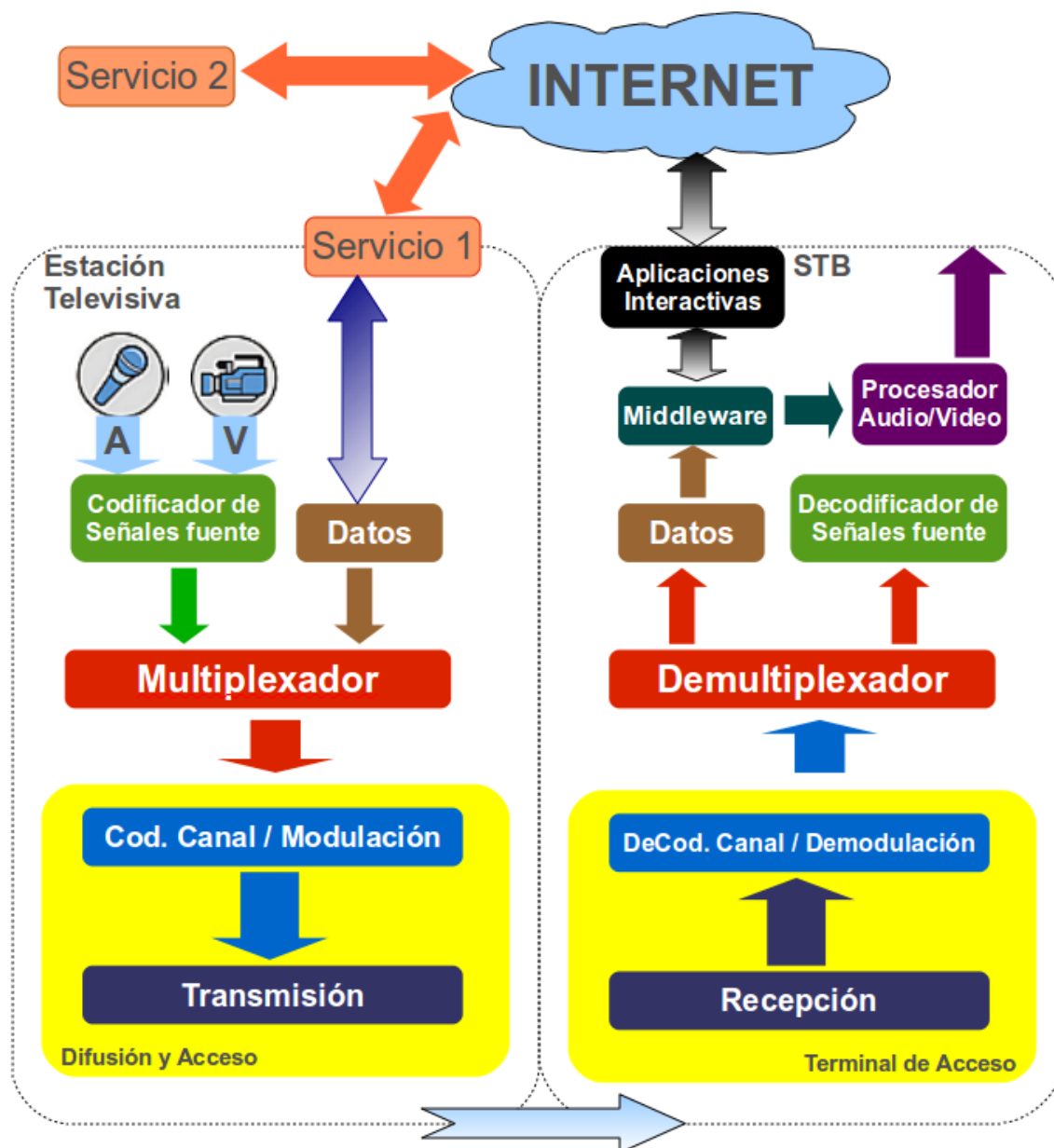
La figura 3-41 muestra todas las etapas de la conversión, transmisión, recepción y métodos de comunicación existentes en el estándar de TVDigital. A partir de la estación televisiva(a la izquierda), video y audio son generados para los programas de televisión (estudio o grabaciones) y son codificadas a los estándares del SBTVD. Los datos que se envían también son generados por el servidor de aplicaciones de la estación. En la fase de multiplexación (transmisión simultánea de dos o más corrientes en un único canal de transmisión), los tres flujos se convierten en uno a fin de proceder con la transmisión de la misma. Sin embargo, antes de la transmisión en sí, el flujo de datos multiplexados necesita ser modulada. La señal modulada se envía a la antena de transmisión para que se propague en el aire. A la derecha se representan las secuencias de los procesos que ocurren dentro de la terminal de acceso (por ejemplo, caja de set-top box). La señal es recibida por una antena de tipo UHF (Ultra High Frequency), se desmodula dividiendo la señal recibida en tres señales nuevas: audio, video y datos.

Figura 3-40: Posibles formas de canales de comunicación



Fuente: PUHLMANN, Christian. Sistema Brasileiro de TV Digital. 2008

Figura 3-41: Sistema de transmisión y recepción del estándar ISDBT-t



Fuente: PUHLMANN, Christian. Sistema Brasileiro de TV Digital. 2008

El audio y video siguen directamente a la pantalla de televisión, mientras que los datos van al middleware localizado en la unidad de procesamiento del equipo receptor y si es necesario se mezcla el audio, el video y la aplicación interactiva.

La aplicación puede requerir la comunicación con cualquier equipo de servicios externos. Para ello, se necesita un canal de comunicación con Internet (la nube).

Entre las diversas formas de comunicación, hasta ahora los más mencionados por los fabricantes, son los siguientes: la línea telefónica (teléfono), red inalámbrica (wi-fi), ADSL (Asymmetric Digital Subscriber Line), PLC (Power Line Communication), GSM/GPRS, CDMA2000/1xRTT y WiMAX.

Una alternativa viable para el canal de retorno es el uso de la tecnología ADSL, que en comparación con años anteriores el costo del servicio se ha reducido, aunque si bien la penetración aún es mínima en relación con el servicio de la televisión abierta, queda en manos de las autoridades, no solo por el hecho de contar con la tecnología que aporte interactividad a las aplicaciones de televisión digital, si no para acercar la tecnología que Internet aporta a cada hogar en cada rincón del Ecuador, es una tarea extremadamente difícil pero no está por demás recordar que hoy por hoy el futuro está instalado con las nuevas tecnologías.

3.7 CONSIDERACIONES FINALES

En este capítulo se destinó al desarrollo de la aplicación interactiva, conocer a fondo los lenguajes de programación y definir el entorno más adecuado para el diseño de estas aplicaciones, como programador pienso que es de vital importancia el detenerse a planear correctamente cual serán las herramientas que nos van a acompañar en el desarrollo, escoger las mejores y las que más se adapten a nuestras necesidades es de vital importancia para afrontar de la mejor manera todo el tiempo y esfuerzo que su diseño demande, así mismo, la metodología de desarrollo implica ese detalle que muchas veces no queremos definir, pero que sin duda permite aclarar el panorama de desarrollo, de la mano con los diferentes diagramas que otorguen a las demás personas la posibilidad de entender y comprender la idea principal de una persona o equipo de desarrollo, todos estos detalles harán que futuros proyectos lleguen a buen puerto.

CAPÍTULO 4. SIMULACIÓN Y PRUEBAS

4.1 INTRODUCCIÓN

Dentro de cualquier desarrollo, la parte quizás olvidada muchas veces, otras tantas ignorada es la fase de pruebas. De ellas depende en gran medida evitar tener que corregir ciertas cosas que surgen cuando la aplicación o sistema ya está en producción y que arreglar ciertos *bugs*⁵⁸ en una etapa tan crítica siempre es un gran riesgo que puede involucrar la caída total del sistema o la pérdida de información importante.

En sistemas tradicionales, aplicaciones web o de escritorio, existen desde hace mucho tiempo ciertos mecanismos y herramientas que permiten llevar acabo la mayoría de pruebas que se necesita para afrontar con firmeza la etapa de salida a producción. Las pruebas se enmarcan en dos grandes grupos, las pruebas funcionales, donde la herramienta comúnmente usada se llama *Casos de prueba* y las pruebas de código donde se cuenta con herramientas como test unitarios, pruebas de stress, integración continua, etc. En el mundo de las aplicaciones para la televisión digital no existe una herramienta específica, realmente porque no hace falta, simplemente se necesita adaptar las ya existentes en un ámbito diferente, teniendo en cuenta que el entorno donde las aplicaciones interactivas se ejecutan son un ambiente más hostil, esto quiere decir que estas aplicaciones se ejecutan en equipos con recursos de hardware mucho menores que una computadora o mucho menos que en un servidor de grandes prestaciones. Además de contar con un hardware limitado, las aplicaciones interactivas tienen que estar en la capacidad de iniciarse automáticamente en caso de que un fallo ocurriere y deben estar a la altura de un usuario que querrá sin demora ver los resultados de su interacción.

Por esta razón las pruebas que se llevarán a acabo principalmente en la aplicación DigitalTv serán tomando en cuenta que se tiene dos lenguajes completamente distintos pero que interactúan entre si, NCL y Lua, en el mundo de

⁵⁸ Palabra inglesa para nombrar los errores de programación que producen fallas en el software en cualquiera de sus etapas de desarrollo.

la televisión digital a esta combinación se la llama aplicaciones NCLua, que desde ahora en adelante se nombra así.

Luego llega la hora de simular la aplicación y ver los resultados, la simulación se hará únicamente a nivel de PC para emular el comportamiento de un receptor de TVD interactivo, con diferentes implementaciones de Ginga que se encuentran disponibles tanto en instalaciones nativas⁵⁹ como en máquinas virtuales y discos Live CD⁶⁰ con Ginga pre instalado.

Finalmente la documentación del código fuente y los manuales de Usuario que serán el soporte importante en tareas de actualización o mantenimiento de la aplicación desarrollada.

4.2 PRUEBAS

4.2.1 PRUEBAS FUNCIONALES

Las pruebas tiene como objetivo verificar que el producto cumpla con todos los requerimientos funcionales establecidos y especificados mediante los casos de uso. Además de reducir, eliminar y prevenir las deficiencias de calidad de los productos a obtener. Para cumplir con este objetivo se realizó un diseño de los casos de prueba que sirvió como guía para ejecutar las pruebas.

4.2.1.1 Casos de prueba

Los casos de uso se consideran la guía para todo el proceso de desarrollo de software, En algunos casos de uso intervienen varios componentes, entonces los casos de prueba permitirán probar tanto la funcionalidad del sistema como la integración de los componentes. Cada caso de prueba se describe utilizando el formato indicado en la tabla 4-1.

⁵⁹ Desde el código fuente se puede crear los binarios o ejecutables de cierta aplicación.

⁶⁰ LiveCd es una herramienta que permite ejecutar una distribución Gnu/Linux desde el Cd-rom sin instalar nada en el equipo huésped ni alterar el disco duro de su máquina.

Tabla 4-1: Formato de descripción de casos de prueba

N. Caso de prueba	Número del caso de prueba
Referente al caso de uso	Nombre del caso de uso
Nombre	Nombre del caso de prueba
Entradas	Interacción recibida por el usuario
Salidas	Respuesta que mostrará el sistema
Descripción	Descripción breve del caso de prueba
Procedimiento de prueba	Secuencia de pasos para realizar la prueba
Resultados esperados	Descripción de comportamiento ideal de la aplicación durante la ejecución del procesamiento de prueba.

4.2.1.1.1 Caso de prueba: Menú principal

La prueba del caso de uso Menú principal se muestra en la tabla 4-2.

Tabla 4-2: Descripción de caso de prueba Menú principal

N. Caso de prueba	1
Referente al caso de uso	Menú principal
Nombre	Acceso al menú principal luego de interacción del usuario
Entradas	Botón “ROJO” control remoto
Salidas	Indicador de interactividad y luego de interacción, Menú principal
Descripción	Aplicación en estado inicial, a la espera de la interacción del usuario.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. La aplicación interactiva se carga junto con el video y audio de la programación. 2. El indicador de interactividad aparece en la pantalla. 3. Se presiona el botón “ROJO” del control remoto. 4. El menú principal aparece, la opción Aliméntate Ecuador queda seleccionada.
Resultados esperados	<ol style="list-style-type: none"> 1. El indicador de interactividad anuncia que la aplicación ha sido exitosamente carga. 2. El menú principal aparece, todas las opciones son mostradas, la gráfica se “dibuja” correctamente, las indicaciones se pueden leer claramente.

4.2.1.1.2 Caso de prueba: Aliméntate Ecuador

Tabla 4-3: Descripción de caso de prueba Aliméntate Ecuador

N. Caso de prueba	2
Referente al caso de uso	Aliméntate Ecuador
Nombre	Ejecución de la aplicación Aliméntate Ecuador
Entradas	Botón “ROJO” control remoto Botón “VERDE” control remoto Botón “AMARILLO” control remoto Botón “AZUL” control remoto Botón “ENTER” control remoto Botón “CURSOR LEFT” control remoto
Salidas	Gráfica, platos, descripción platos legible.
Descripción	La aplicación muestra 3 platos de comida, el usuario escoge uno de ellos y revisa calidad de nutrientes del plato seleccionado.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. Seleccionar opción Aliméntate Ecuador, acceder. 2. Se presiona el botón “VERDE” del control remoto y se muestra la información correspondiente a este plato. 3. Se presiona el botón “CURSOR LEFT” para regresar al menú de platos. 4. Se presiona el botón “AMARILLO” del control remoto y se muestra la información correspondiente a este plato. 5. Se presiona el botón “CURSOR LEFT” para regresar al menú de platos. 6. Se presiona el botón “AZUL” del control remoto y se muestra la información correspondiente a este plato. 7. Se presiona el botón “ROJO” para finalizar la aplicación Aliméntate Ecuador y regresar al menú principal.
Resultados esperados	<ol style="list-style-type: none"> 1. La aplicación muestra el menú de platos, a cada interacción se muestra el plato escogido y su información, clara y legible. Retorna al menú principal exitosamente.

4.2.1.1.3 Caso de prueba: Historia del Ecuador

Tabla 4-4: Descripción de caso de prueba Historia del Ecuador

N. Caso de prueba	3
Referente al caso de uso	Historia del Ecuador
Nombre	Ejecución de la aplicación Historia del Ecuador
Entradas	Botón “ROJO” control remoto Botón “CURSOR LEFT” control remoto Botón “CURSOR RIGTH” control remoto Botón “ENTER” control remoto Botones “1”, “2”, “3”, “4” control remoto
Salidas	Gráfica, descripción, preguntas, respuestas legibles
Descripción	La aplicación muestra las preguntas con sus posibles respuestas, al final muestra el porcentaje de aciertos y de errores.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. Seleccionar opción Historia del Ecuador, acceder. 2. Leer atentamente primera pregunta, presionar botón “3”, para continuar a la siguiente pregunta presionar botón “CURSOR RIGTH”. 3. Leer atentamente segunda pregunta, presionar botón “1”, para continuar a la siguiente pregunta presionar botón “CURSOR RIGTH”. 4. Leer atentamente tercera pregunta, presionar botón “1”, para continuar a la siguiente pregunta presionar botón “CURSOR RIGTH”. 5. Presionar botón “CURSOR LEFT” para regresar a la pregunta 6. Presionar botón 2 para corregir selección anterior 7. Presionar el botón “ENTER” para verificar respuestas, aprender de opciones equivocadas. 8. Se presiona el botón “ROJO” para finalizar la aplicación Historia del Ecuador y regresar al menú principal.
Resultados esperados	<ol style="list-style-type: none"> 1. El indicador de interactividad anuncia que la aplicación ha sido exitosamente carga. 2. El menú principal aparece, todas las opciones son mostradas, la gráfica se “dibuja” correctamente, las indicaciones se pueden leer claramente.

4.2.1.1.4 Caso de prueba: Entidades Públicas

Tabla 4-5: Descripción de caso de prueba Entidades Públicas

N. Caso de prueba	4
Referente al caso de uso	Entidades Públicas
Nombre	Ejecución de la aplicación Entidades Públicas
Entradas	Botón “ROJO” control remoto Botón “CURSOR LEFT” control remoto Botón “ENTER” control remoto Botones “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “0” control remoto
Salidas	Gráfica, descripción, campo cédula, resultados búsqueda.
Descripción	La aplicación permite el ingreso de el número de identificación y verifica si tiene un turno registrado en el Sistema de Actas de Finiquito.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. Seleccionar opción Entidades Públicas, acceder. 2. Seleccionar el botón “ENTER”, verificar que la búsqueda no se ha realizado 3. Ingresar los números 12345, seleccionar el botón “ENTER”, tampoco se realiza la búsqueda. 4. Seleccionar el botón “CURSOR LEFT” para borrar todos los números ingresados en el punto anterior. 5. Ingresar el número de identificación 1715835524 usando el teclado numérico del control remoto, se visualiza el mensaje que para continuar seleccionar “OK/Enter”. 6. Seleccionar el botón “ENTER”, la búsqueda se realizó, y se visualiza los resultados. 7. Seleccionar el botón “CURSOR LEFT” para regresar al ingreso, primera pantalla. 8. Se presiona el botón “ROJO” para finalizar la aplicación Entidades Públicas y regresar al menú principal.
Resultados esperados	<ol style="list-style-type: none"> 1. El indicador de interactividad anuncia que la aplicación ha sido exitosamente cargada. 2. El menú principal aparece, todas las opciones son mostradas, la gráfica se “dibuja” el resultado de la búsqueda funciona perfectamente, los accesos con el teclado operan normalmente.

4.2.1.1.5 Caso de prueba: GamaTv

Tabla 4-6: Descripción de caso de prueba GamaTv

N. Caso de prueba	5
Referente al caso de uso	GamaTv
Nombre	Ejecución de la aplicación GamaTv
Entradas	Botón “ROJO” control remoto Botón “ENTER” control remoto Botón “CURSOR LEFT” control remoto Botón “CURSOR RIGTH” control remoto
Salidas	Gráfica, tweets de GamaTv
Descripción	La aplicación muestra la última información de la cuenta de Twitter de GamaTv
Procedimiento de prueba	<ol style="list-style-type: none"> 1. Seleccionar opción GamaTv, acceder. 2. Al ejecutar la aplicación la conexión con los servidores de Twitter se realiza, dependiendo del canal de retorno esta puede demorar unos minutos. 3. Se presiona el botón “CURSOR LEFT” para acceder a mensajes más antiguos. 4. Se presiona el botón “ROJO” para finalizar la aplicación GamaTv y regresar al menú principal.
Resultados esperados	1. La aplicación muestra los últimos Tweets de GamaTv, conexión a Internet es requerido. Retorna al menú principal exitosamente.

4.2.1.1.6 Caso de prueba: Ups

Tabla 4-7: Descripción de caso de prueba Ups

N. Caso de prueba	6
Referente al caso de uso	Ups
Nombre	Ejecución de la aplicación Ups
Entradas	Botón "CURSOR UP" control remoto Botón "CURSOR DOWN" control remoto Botón "ENTER" control remoto Botón "ROJO" control remoto
Salidas	Gráfica, segundo menú, Misión y Visión
Descripción	La aplicación muestra un segundo menú para acceder a la Misión, Visión o regresar al menú principal.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. Seleccionar opción Ups, acceder. 2. Escoger del segundo menú la opción Misión con el botón "ENTER" del control remoto. 2. Se presiona el botón "ROJO" del control remoto para regresar al segundo menú. 3. Se presiona el botón "CURSOR DOWN" para seleccionar la opción Visión , luego el botón "ENTER" para acceder. 2. Se presiona el botón "ROJO" del control remoto para regresar al segundo menú. 3. Se presiona el botón "CURSOR DOWN" para seleccionar la opción Regresar , luego el botón "ENTER" para ocultar segundo menú y dar paso al menú principal. 2. Se presiona el botón "CURSO DOWN" para seleccionar la opción Salir, el menú principal desaparece.
Resultados esperados	1. La aplicación el segundo menú, se puede acceder al mensaje de Misión y Visión, se puede retornar al segundo menú y también retornar al menú principal. La opción "Salir" cierra el menú principal y muestra el indicador de interactividad, la aplicación queda en estado inicial, lista para repetir al rutina.

4.3 SIMULACIÓN

El objetivo es instalar Ginga en la PC para disponer de un entorno similar al existente en un Set Top Box con Ginga, para poder probar la aplicación interactiva.

Ginga al ser software libre tiene a disposición de todos su código fuente, es posible descargar todos los archivos necesarios para realizar una instalación nativa⁶¹ en cualquier distribución Gnu/Linux como en este caso.

Existe también disponibles otro tipo de implementaciones de Ginga, máquinas virtuales, LiveCds, paquetes *deb*⁶², etc.

4.3.1 EMULADOR GINGA-NCL

El emulador Ginga-NCL (Figura 4-1) permiten simular un set top box. Para la ejecución del emulador es necesario tener instalada la Máquina Virtual de Java (JVM).

Esta herramienta muestra todos los errores en caso de existir, a través de la consola (tanto en ambiente GNU/Linux, Windows y MacOS), además esta herramienta posee un control remoto interactivo que puede ser utilizado a través del mouse.

4.3.2 INSTALACIÓN NATIVA

Una instalación nativa permite ejecutar Ginga en el sistema operativo existente, interactuando con él como con cualquier otra aplicación existente en la PC.

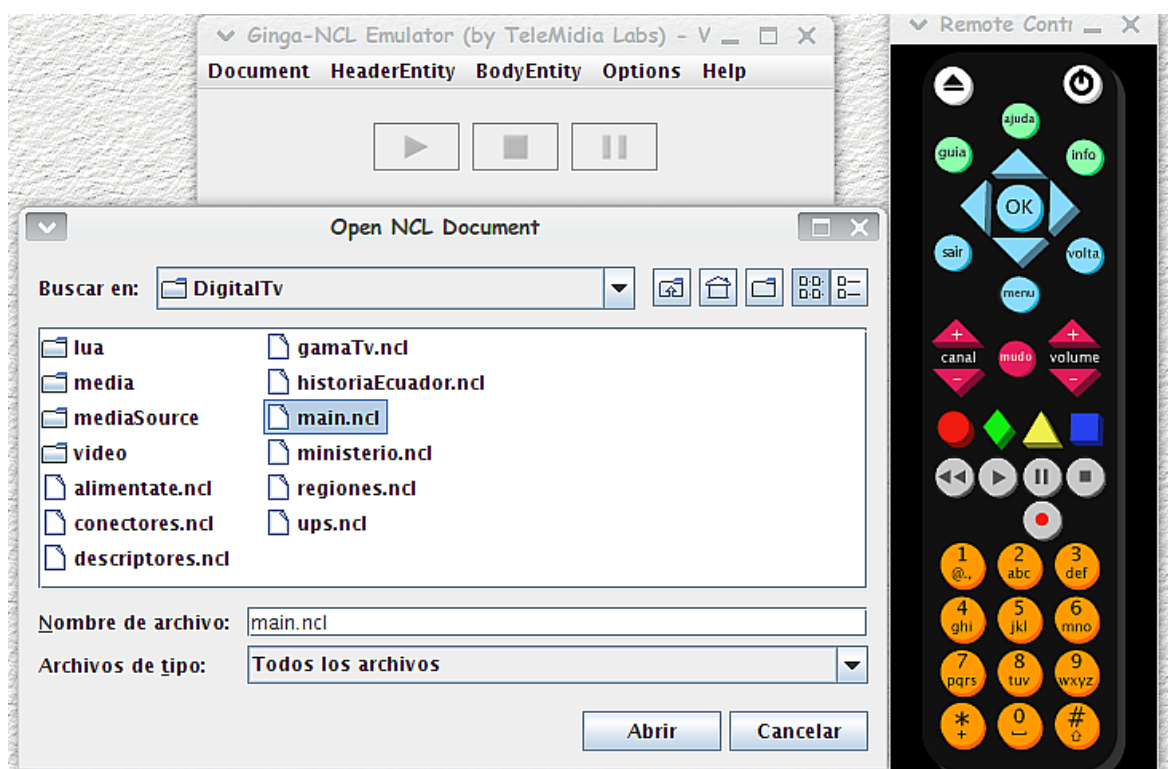
La ventaja de trabajar con una instalación nativa es la simplicidad en el manejo de archivos y se mejora la velocidad de la PC. La desventaja es que el proceso de instalación no es sencillo ni está automatizado aún.

61 Para Ginga existe documentación oficial para realizar una instalación desde el código fuente solo en distribuciones Gnu/Linux.

62 Paquetes ejecutables para distribuciones Gnu/Linux basadas en Debian.

La complejidad en este proceso de instalación aleja a los desarrolladores a inclinarse por esta opción, pero cuando la agilidad de desarrollo se trata sin duda esta es la mejor opción, ya que se puede programar la aplicación, guardar los cambios y simplemente con una línea de comando visualizar la simulación inmediatamente.

Figura 4-1: Emulador de Ginga-NCL. TeleMidia



Fuente: El autor

Las instrucciones para llevar a cabo este tipo de instalación se encuentran en el Anexo D. Esta información fue tomada de la documentación oficial de los miembros de desarrollo del laboratorio TeleMidia de la PUC-Rio, a partir de esta implementación ha surgido otra documentación también importante del departamento de investigación de la Universidad de la Plata, este laboratorio de investigación se llama Lifa, ellos hicieron cambios al código fuente, mejoras, arreglos de *bugs* e *issues*⁶³ aunque estos aún no están incorporados al Ginga

⁶³ Un issue puede ser el arreglo de un fallo, una característica pedida, una tarea, un pedido de Documentación específico y todo tipo de solicitud al equipo de desarrollo.

original se espera como es el espíritu del software libre que estos sean acogidos por la PUC-Rio. De esta manera también es posible instalar esta versión de Ginga, ya que, su código fuente también es de acceso público⁶⁴.

En los dos casos, el Ginga instalado solo tiene soporte para aplicaciones NCLua, si se ha desarrollado aplicaciones con el API de Java esta implementación de Ginga no es una opción.

Para dar soporte a las aplicaciones escritas para Ginga-J se tiene la implementación OpenGinga, desarrollada por la Universidad de Paraiba, específicamente el laboratorio de investigación Lavid. Esta implementación incluye el Ginga-NCL y el Ginga-J y además agrega ciertas opciones adicionales como un generador de programación, sintonizador de canales, archivos de configuración, etc. El código fuente también disponible para su descarga⁶⁵ y las instrucciones de instalación que se puede observar en el Anexo D.

OpenGinga es la implementación más completa y la que más se acerca a la implementación de un Set Top Box.

4.3.3 MÁQUINAS VIRTUALES

A través de un software de virtualización, se pueden tener en un sistema operativo máquinas virtuales ejecutando otro sistema operativo. Se habla de un sistema operativo anfitrión (host) y otro huésped (guest).

Esta modalidad permite que un usuario de cualquier sistema operativo, Windows, Gnu/Linux, MacOS, pueda ejecutar “dentro” de su propio sistema otro nuevo sistema (en este caso algún Gnu/Linux con Ginga pre-instalado).

La ventaja de trabajar con máquinas virtuales es la facilidad de disponer de una implementación de Ginga rápidamente, aunque para desarrollar puede ser más engorroso por el hecho de contar con dos ambientes simultáneos (el host y el

64 http://svn.softwarepublico.gov.br/trac/ginga/wiki/Building_Wiki_GingaNCL

65 <http://gingacdn.lavid.ufpb.br/attachments/download/363/OpenGinga-Source-0.4.3.tar.gz>

quest) entre los que transferir archivos, etc. Para usuarios de Windows, es la única opción disponible hasta ahora.

A la fecha, se han publicado máquinas virtuales Ginga para las siguientes opciones de software de virtualización: VirtualBox o VMware. Es necesario contar con una instalación de VirtualBox o VMware para poder levantar alguna de las máquinas virtuales disponibles.

Los pasos a seguir para utilizar una máquina virtual Ginga son:

- Instalar software de virtualización (VirtualBox o VMware).
- Descargar una máquina virtual con Ginga.
- Levantar la máquina virtual desde el software de virtualización.
- Probar que la implementación de Ginga funcione en la máquina virtual.

4.3.3.1 VirtualBox

Para el software de virtualización VirtualBox existe una implementación realizada por el laboratorio Lifia con la versión Ginga 1.2(figura 4-2), versión que fue modificada, de la versión de Ginga de la Puc-Rio, pero que estos cambios o mejoras no afectan en nada el comportamiento de la aplicación DigitalTv.

4.3.3.1.1 Instalar software de virtualización

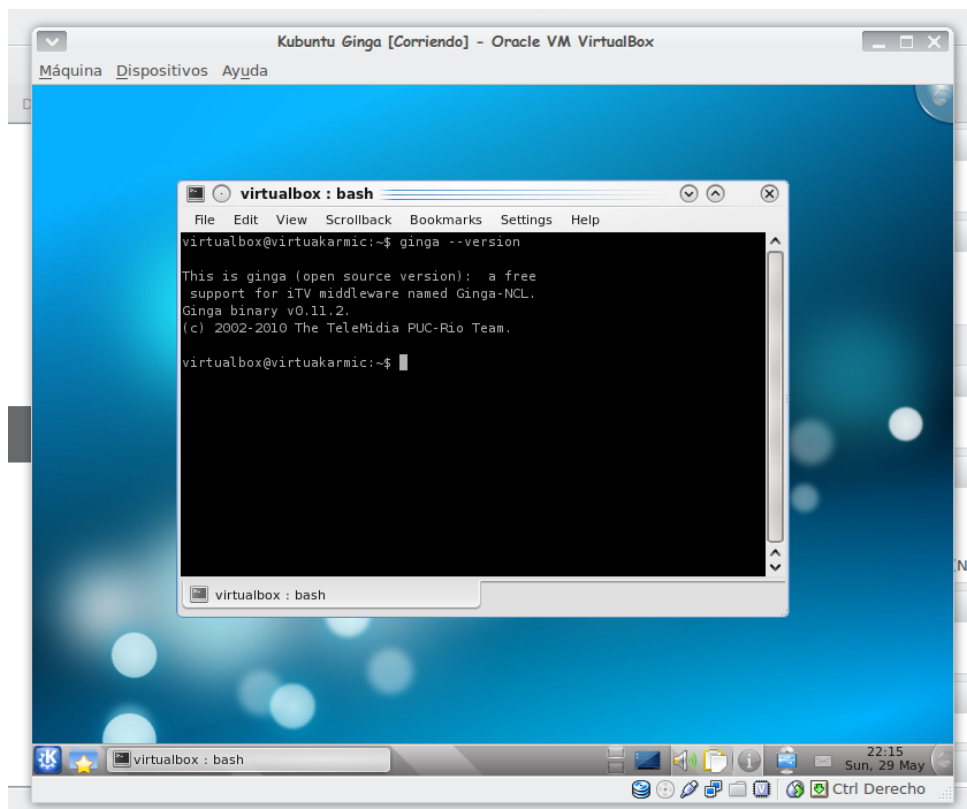
- Se debe descargar y seguir las instrucciones de instalación del sitio de VirtualBox⁶⁶. VirtualBox es software libre y está disponible para su descarga directa para Microsoft Windows, Gnu/Linux o MacOS.
- Se debe descargar la imagen de disco de los repositorios de Lifia⁶⁷

⁶⁶ <http://www.virtualbox.org/wiki/Downloads>

⁶⁷ <ftp://tvd.lifia.info.unlp.edu.ar/SetTopBoxVirtual/Kubuntu-Karmic-i386-GingaLifia.vdi.tgz>

- Descomprimir el archivo `.vdi` que contiene la imagen con Kubuntu⁶⁸ y Ginga-NCL.

Figura 4-2: Kubuntu con implementación de Ginga de Lifa en VirtualBox



Fuente: El autor

4.3.3.2 VMware

Para el software de virtualización VMware existen dos implementaciones realizadas, una por TeleMidia y otra por el Lifa.

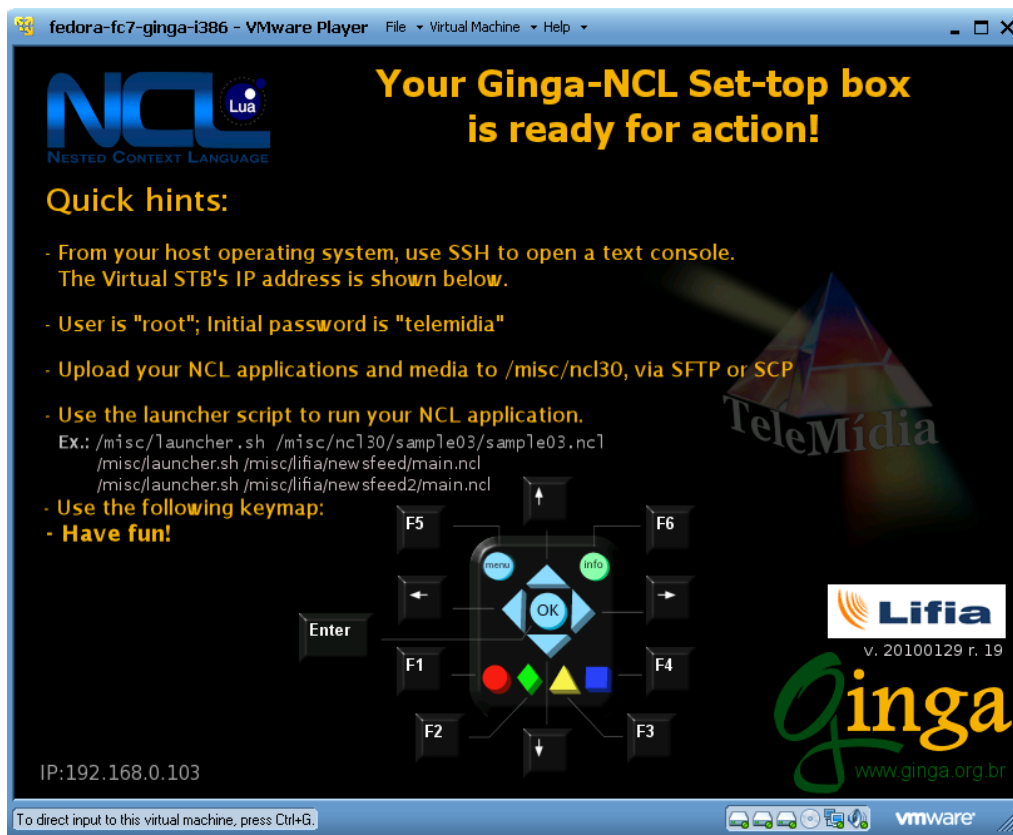
4.3.3.2.1 Instalar software de virtualización

- Se debe descargar y seguir las instrucciones de instalación del sitio de VMware⁶⁹. VMWPlayer es una versión gratuita de este software, la versión comercial tiene un costo por licencia. Está disponible para Microsoft Windows, Gnu/Linux o MacOS.

⁶⁸ Distribución Gnu/Linux derivada de Ubuntu con entorno gráfico KDE.

⁶⁹ <http://downloads.vmware.com/d/>

Figura 4-3: Fedora con implementación de Ginga de Lifia en Vmware



Fuente: El autor

- Se debe descargar la imagen de disco de los repositorios de Lifia⁷⁰ y también se puede descargar la imagen de disco de la implementación de TeleMidia⁷¹.
- Descomprimir el archivo .vmx que contiene la imagen, bien sea de la implementación del Lifia que usa una distribución Fedora(Figura 4-3) o la implementación de TeleMidia que usa una distribución Ubuntu.

En el caso de las máquinas virtuales es necesario establecer una conexión ssh o ftp para enviar los archivos de la aplicación desde la máquina de desarrollo hacia la máquina virtualizada, así mismo los requisitos de hardware de la máquina sobre la cual se va a virtualizar son las siguientes:

- Pentium 4 3.0GHz o superior. HyperThreading recomendado. Doble núcleo

70 <ftp://72.14.182.9/SetTopBoxVirtual/fedora-fc7-ginga-i386-lifia-20100218-dev.tar.gz>

71 <http://www.ncl.org.br/ferramentas/ubuntu-server10.10-ginga-i386.zip>

ideal.

- Memoria RAM de 1 Gb o más, 2 Gb recomendado.
- Placa de Vídeo con 64Mb o superior. Chipsets nVidia o ATI.
- Disco duro con 5Gb de espacio libre.
- Placa de sonido.

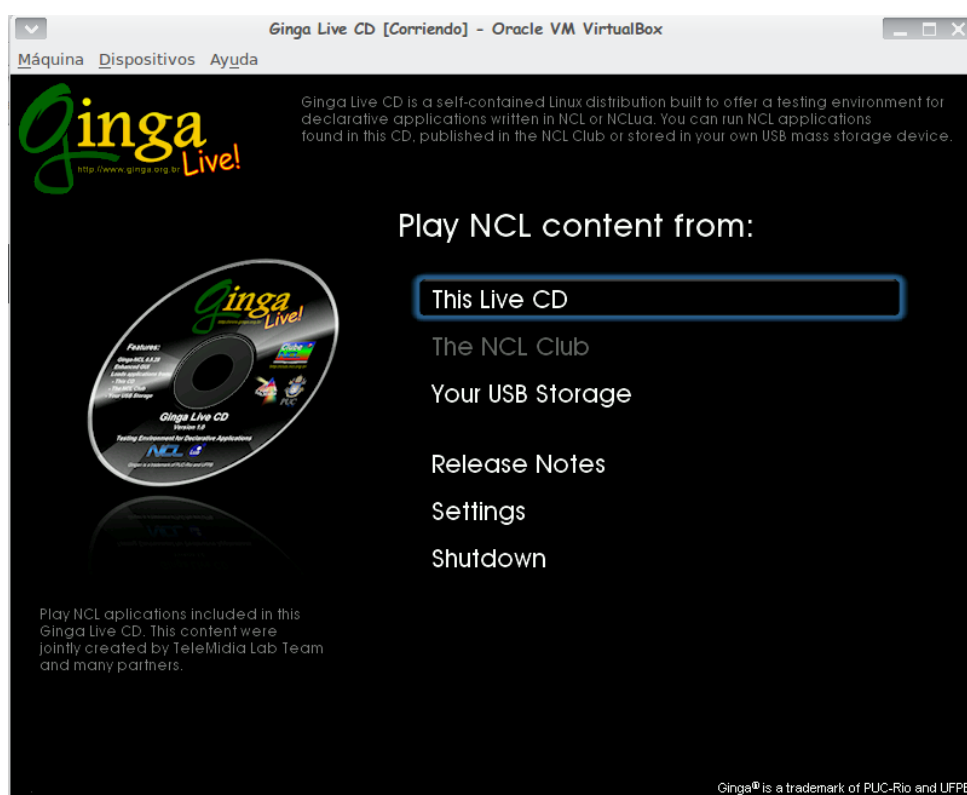
4.3.4 PAQUETES DEB

La ventaja de disponer del código fuente de Ginga es que es posible a partir de ellos crear archivos binarios para facilitar su instalación, este es el caso de la comunidad Ginga Argentina que pone a disposición los binarios *.deb*, archivos que en distribuciones basadas en Debian se instalan de forma simple y rápida, el inconveniente de esta opción es que a veces no todos los archivos necesarios se pueden empaquetar, por lo que es muy probable que al intentar ejecutarlos el sistema operativo no tenga las dependencias satisfechas, es decir la instalación fallará, en Debian existe un administrador para estos paquetes que permiten ejecutar el archivo y comprobar si todos los archivos necesarios se encuentran instalados, si no es así muestra una lista de las aplicaciones faltantes y no se podrá ejecutar la instalación correctamente hasta que estos estén instalados.

4.3.5 LIVECD

El LiveCd con la implementación de Ginga fue creado por el laboratorio de investigación TeleMidia, es un CD de arranque que se ejecuta al momento de colocarlo en la bandeja del CD-ROM de la máquina.

Figura 4-4: Live Cd implementación de Ginga



Fuente: El autor

Esta implementación difiere en ciertos aspectos con la máquina virtual que el mismo laboratorio elaboró, en la siguiente tabla 4-9 se aprecia una comparación entre estas dos implementaciones.

Entre las opciones más importantes que posee esta implementación es el recurso para montar dispositivos de almacenamiento externo y buscar entre sus documentos, archivos con extensión NCL y ponerlos a disposición para su ejecución. Además si se detecta conexión a Internet, el Live CD es capaz de conectarse con un repositorio de aplicaciones NCL, descargarlo y ejecutarlo.

Club NCL se llama el gran repositorio de aplicaciones interactivas escritas en NCLua, todos los trabajos ahí hospedados tienen licencia Creative Commons, están disponibles para su estudio y prueba.

Tabla 4-8: Comparativa Implementaciones Ginga

	Implementaciones Ginga		
	Telemidia	Lavid	Lifia
Instalación nativa	Si	Si	Si
Máquina virtual	Si	Si	Si
LiveCd	Si	No	No
Ginga-NCL	Si	Si	Si
Ginga-J	No	Si	No
Funciones adicionales. EPG,CloseCaption,etc	No	Si	No
Implementación en Set Top Box	Si - CORADIR CDR 1000 - Newtronic DV-5306	No	Si - STB UTE 740

En la tabla 4-8 se puede apreciar una comparativa entre las diferentes implementaciones con las que se cuentan en este momento, este análisis determina que herramienta se usará para la simulación de la aplicación interactiva. A continuación las razones por las cuales se ha decidido usar la implementación de Lavid(OpenGinga) para la simulación.

- Las tres implementaciones tienen soporte para el componente Ginga_NCL y Lua. OpenGinga usa la implementación realizada por TeleMidia para la parte descriptiva del middleware(Ginga-NCL).
- Aunque OpenGinga no este presente aún en ningún Set Top Box, Ginga-NCL está comprobado que funciona plenamente en equipos puestos en el mercado Brasileño y Argentino.
- De todas las implementaciones OpenGinga es la única que cuenta con el componente Ginga-J, esto más que una necesidad es un agregado, ya que, la aplicación interactiva desarrollada no hace uso del lenguaje Java.

Tabla 4-9: Comparativa, funcionalidad y características entre Ginga Live CD y Ginga VMware

Características	Ginga Live CD	Ginga VMware
No es necesario instalar el sistema operativo Gnu/Linux	Si	Si
Ginga NCL pre instalado	Si	Si
Versión de Ginga-NCL	0.10.1	0.12.1
Pre requisitos de hardware	- CPU Pentium 4 2.4 Ghz - CD-ROM(48x deseable) - 1 GB de RAM	- CPU Pentium 4 3,0 Ghz - 2 GB de RAM recomendado - Disco Duro 5 GB de espacio
Pre requisitos de software		VMware Player necesario
Modo de operación principal	Interfaz Gráfica de Usuario	Acceso remoto SSH
Modo de operación secundario	Acceso remoto SSH	
Objetivos de uso	- Navegación en aplicaciones NCL. - Ambiente de pruebas de aplicaciones NCLua	Ambiente de pruebas de aplicaciones NCLua
Portabilidad	Excelente. Ideal para el desarrollador que tiene que hacer presentaciones y demostraciones de sus productos sin tener que preocuparse acerca del software o de la máquina .	Buena. Entre las máquinas con gran potencia de procesamiento y el software de virtualización instalado.
Fuerza	Excelente. Bajo consumo de recursos de la máquina, el sistema más estable . Resistente a los problemas con archivos del sistema y configuración errónea.	Buena. Capacidad de re iniciación en caso de problemas de procesamiento.
Resolución	800 X 600	Configurable entre 4:3 o 16:9
Evolución del sistema	Limitada, los cambios en los archivos del sistema no son persistentes ante un reinicio del Live CD	Completo, los cambios quedan registrados

- Las tres herramientas se encuentran implementadas en máquinas virtuales para facilitar el levantamiento de un entorno de simulación, aunque es desaconsejable totalmente en entornos de desarrollo de aplicaciones por el tiempo que implica el copiar los cambios a la máquina virtual y volver a ponerla en marcha. Si se opta por una instalación nativa, OpenGinga es la más fácil de llevar a cabo, por las herramientas proporcionadas que facilitan su instalación, como se muestra en el Anexo E.

OpenGinga ofrece la ventaja de ser la implementación más completa, más fácil de instalar en forma nativa y por ende en etapas de desarrollo una buena alternativa, en etapas de simulación es la mejor herramienta.

4.4 RESULTADOS

Las figuras 4-5 a la 4-15 muestran capturas de pantalla, el resultado del desarrollo de la aplicación interactiva.

Figura 4-5: Menú principal. Aliméntate Ecuador Seleccionada.



Fuente: El autor

Figura 4-6: Aplicación Aliméntate Ecuador



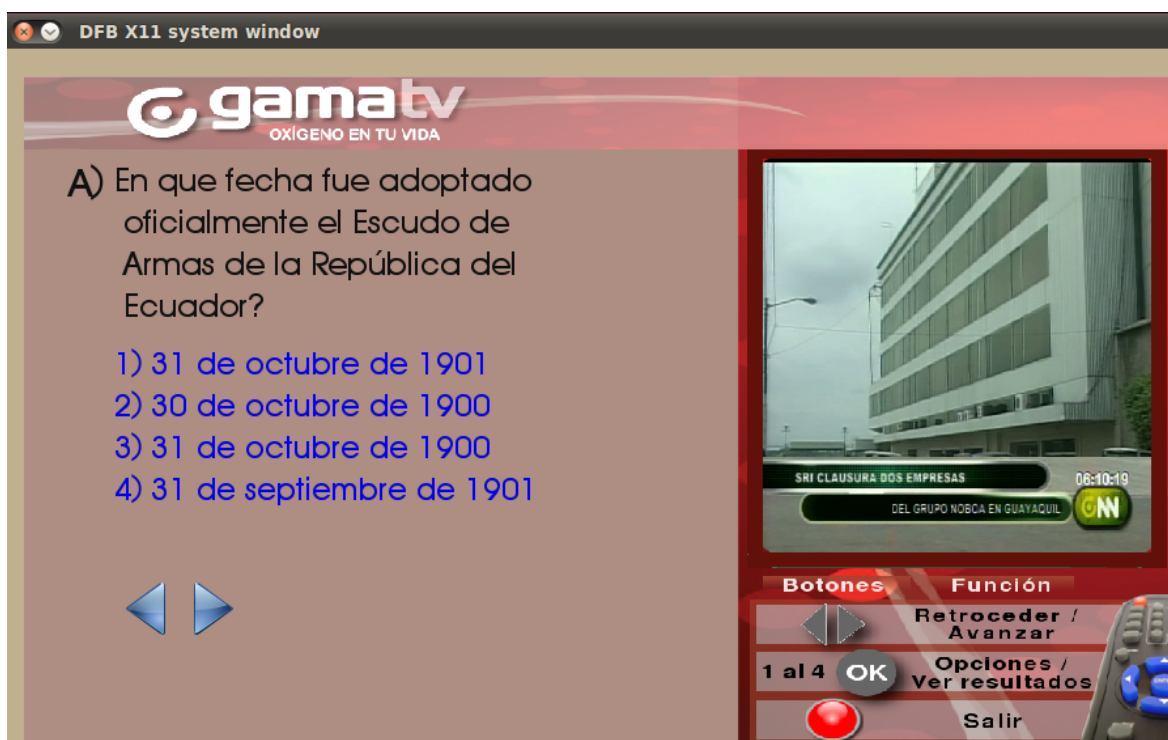
Fuente: El autor

Figura 4-7: Aplicación Aliméntate Ecuador, plato seleccionado.



Fuente: El autor

Figura 4-8: Historia del Ecuador, primera pregunta



Fuente: El autor

Figura 4-9: Historia del Ecuador, resultados, aciertos y errores



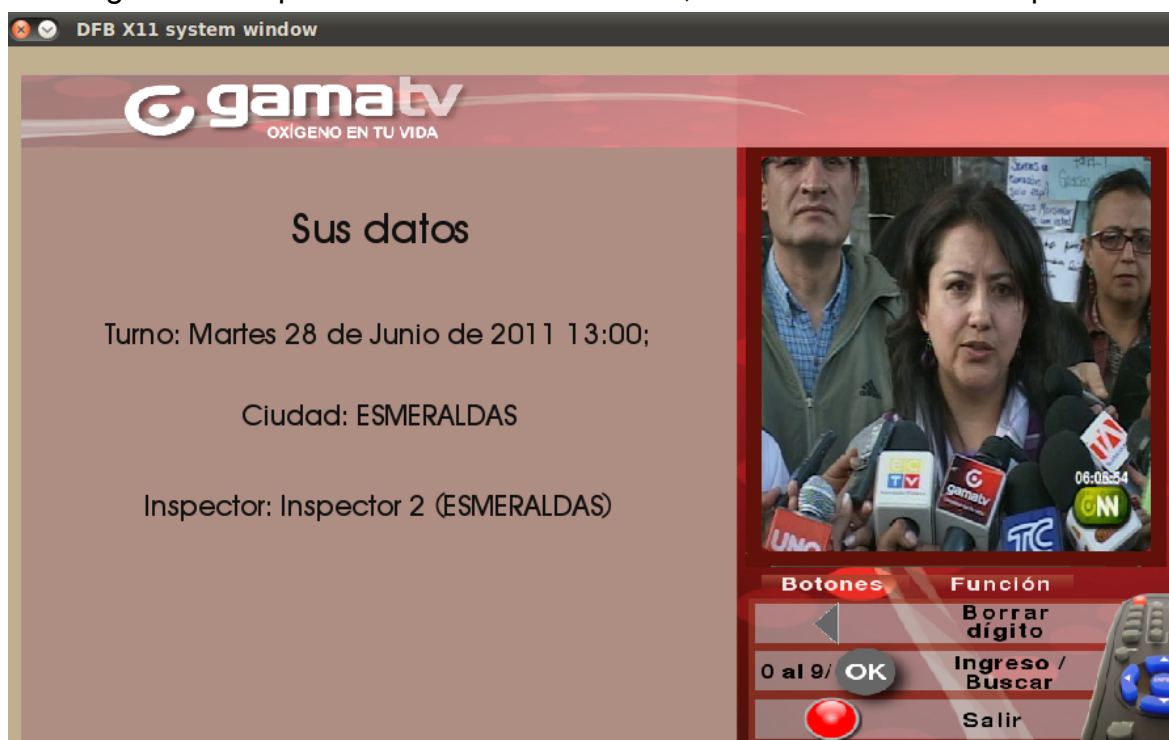
Fuente: El autor

Figura 4-10: Entidades Públicas, Web Service Actas de Finiquito del MRL



Fuente: El autor

Figura 4-11: Aplicación Entidades Públicas, visualización de la búsqueda



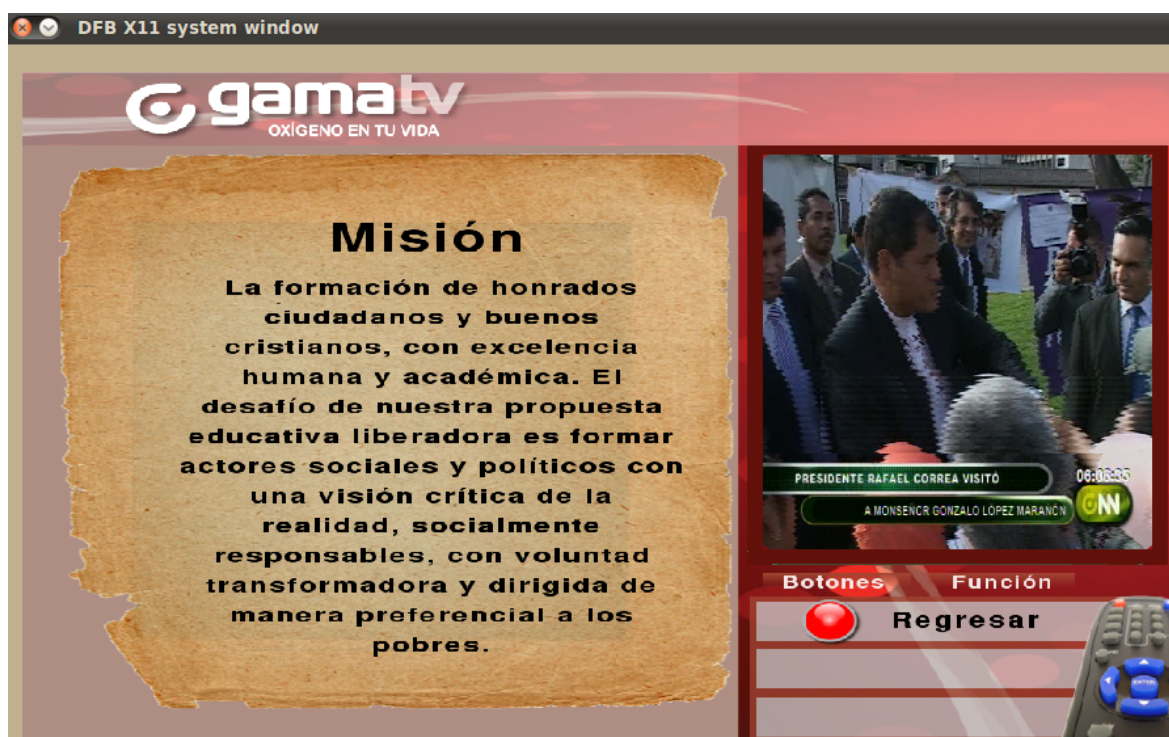
Fuente: El autor

Figura 4-12: Menú opciones Universidad Politécnica Salesiana



Fuente: El autor

Figura 4-13: Universidad Politécnica Salesiana, misión



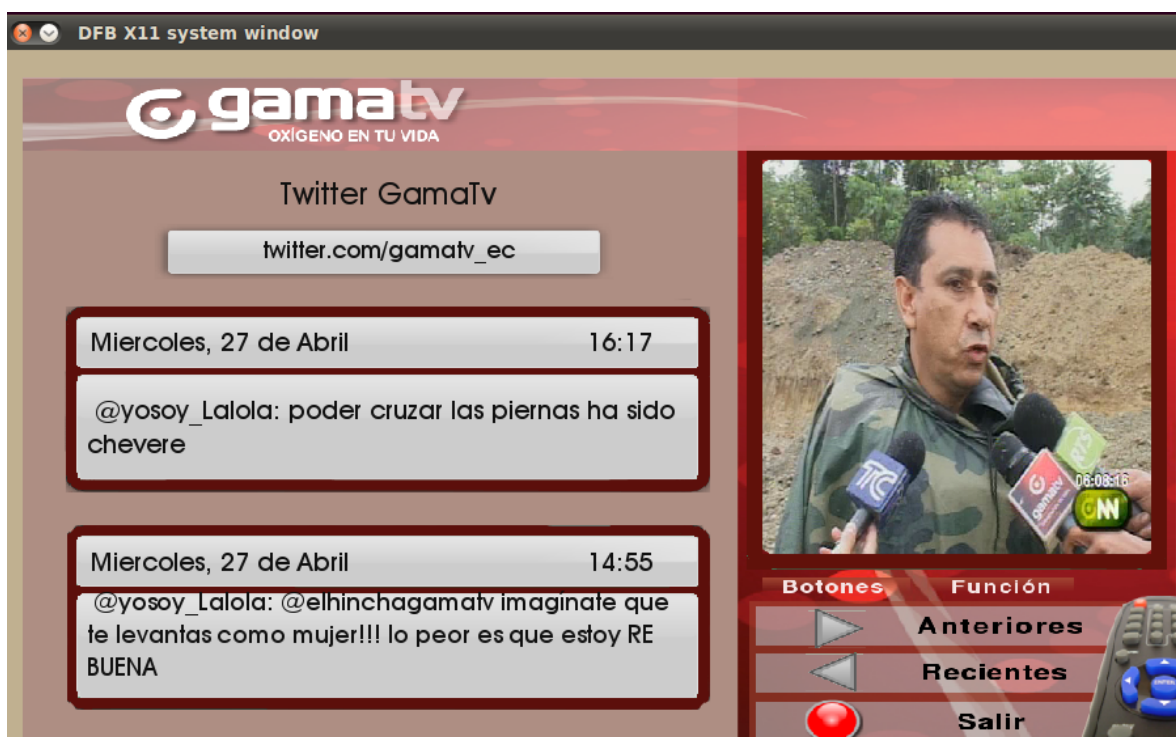
Fuente: El autor

Figura 4-14: Universidad Politécnica Salesiana, visión



Fuente: El autor

Figura 4-15: GamaTV, interactividad con el servicio twitter.



Fuente: El autor

4.5 CONTROL REMOTO

En la norma se establece ciertas condiciones y parámetros para los equipos y componentes, en este caso el control remoto también necesita compartir cierta igualdad entre los diferentes fabricantes.

Dentro de los principales usos del control remoto son acceso a la información sobre programas y servicios, e interactividad (ver Figura 4-16).

4.5.1 FUNCIONES NUMÉRICAS

Los botones numéricos de un control remoto cumple con las siguientes funciones:

- Selección de canales.
- Función de entrada de texto o específica para aplicaciones Ginga.
- Otras funciones dentro de menús propios del receptor.

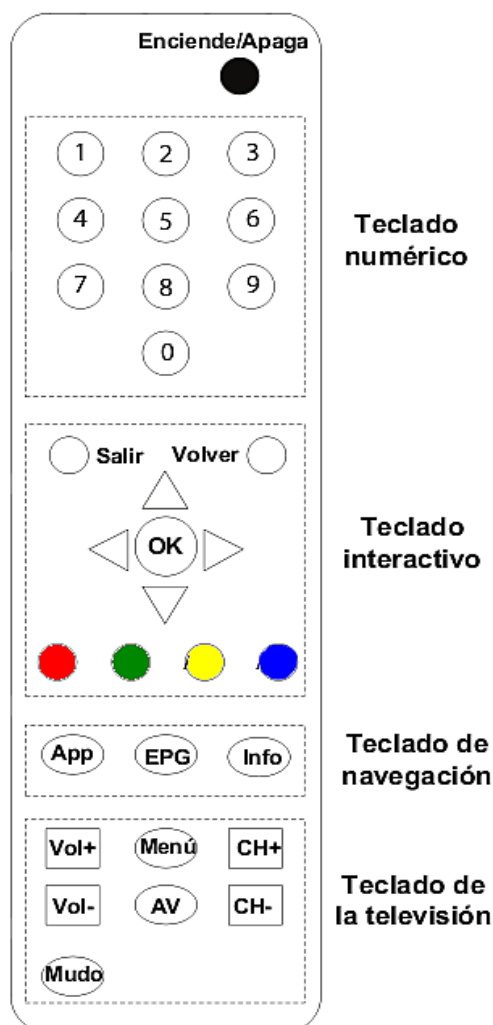
Los ambientes Ginga-NCL y Ginga-J permiten a una aplicación requerir el acceso al grupo de teclado de funciones numéricas de forma dinámica y flexible basada en el contexto de uso.

4.5.2 FUNCIONES INTERACTIVAS

Los botones de funciones de un control remoto cumple con las siguientes funciones:

- Navegar en cualquier aplicación residente del receptor.
- Navegar en cualquier aplicación Ginga.

Figura 4-16: Control Remoto típico para según la norma brasileña



Fuente: ABNT NBR 15606-1 . Codificación de datos y especificaciones de transmisión para radiodifusión digital . 2010

4.5.3 BOTONES DE COLORES (ROJO, VERDE, AMARILLO Y AZUL)

Los botones de colores de un control remoto cumple con las siguientes funciones:

- "CURSOR LEFT"(flecha a la izquierda).
- "CURSOR RIGTH"(flecha a la derecha).
- "CURSOR UP"(flecha hacia arriba).

- “CURSOR DOWN”(flecha hacia abajo).
- Subgrupo de funciones de selección (“ENTER”, Volver y Salir).

El orden de los botones de color debe ser estrictamente cumplido (rojo, verde, amarillo y azul).

Tabla 4-10: Especificaciones para el control remoto

B. Control remoto (funciones mínimas)	Opcional
Botones básicos (encendido / apagado, Números, selección de canal secuencial,)	Obligatorio
Botones de control de volumen	Opcional
Botón de guía de Programación	Opcional
C. Control Remoto (funciones interactivas)	Opcional
Botón de confirmación(Enter, Ok)	Obligatorio
Botón Salir	Obligatorio
Botón Volver(Atrás, Regresar)	Obligatorio
Botones de dirección (▲, ▼, ◀, ▶)	Obligatorio
Botones de colores	Obligatorio

4.6 ESQUEMA DE TELEVISIÓN DIGITAL INTERACTIVA PARA GAMATV.

Cuando las aplicaciones interactivas son emitidas junto a la señal de audio y video, dentro de la estación televisiva se necesita agregar un componente extra que permita que en la señal que entregan a los televidentes vaya incluida la interactividad.

En la figura 4-17 se puede observar un esquema básico pero completo para emisiones de televisión digital interactiva, de donde es posible dividirla en tres aspectos importantes:

- Bloque de emisión(Broadcast Head-end).

- Bloque de edición y desarrollo(Authoring Studio).
- Bloque de pruebas de aplicaciones(Application Testing Enviroment).

4.6.1 BLOQUE DE EMISIÓN

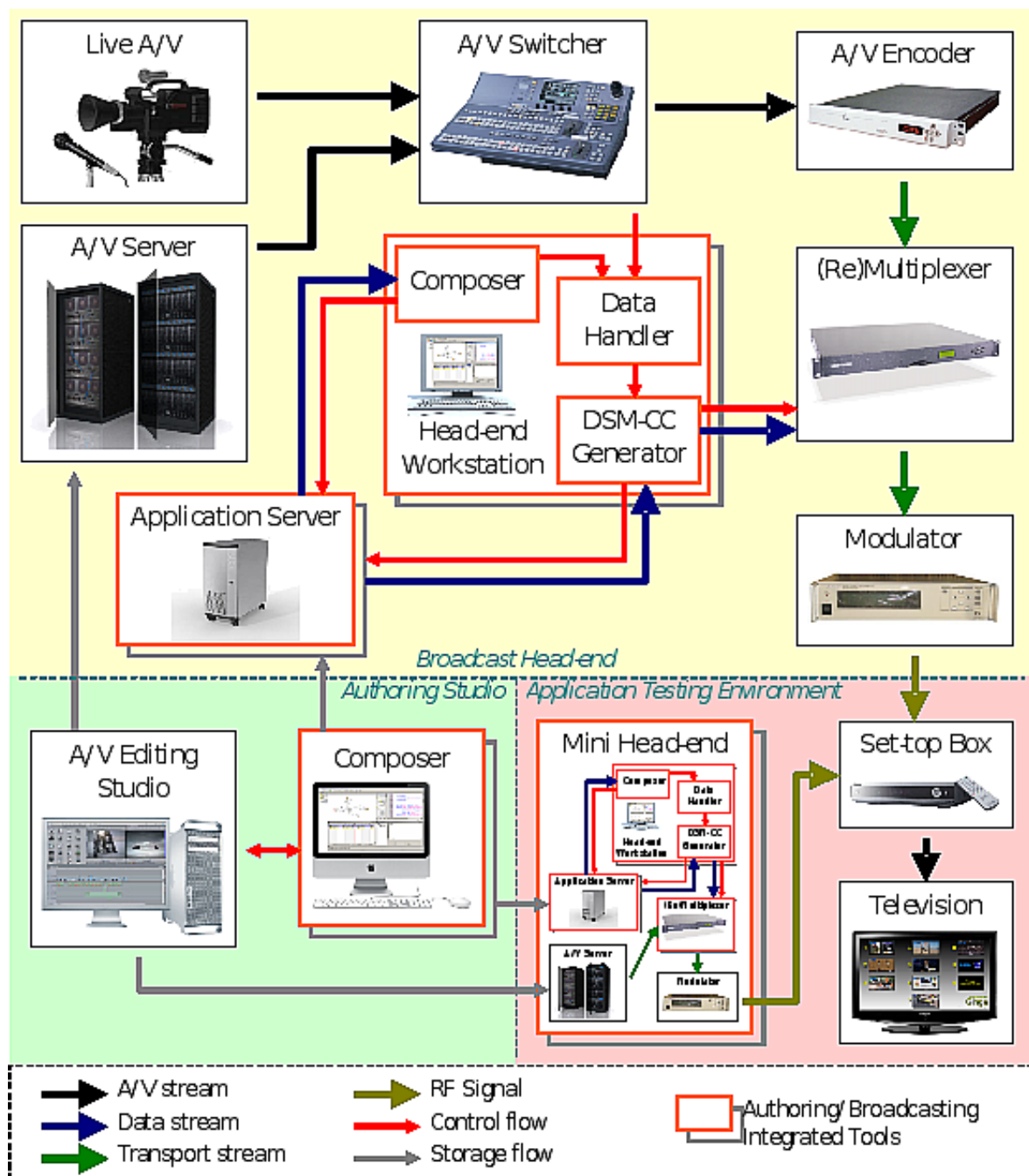
El bloque de emisión se puede representar en un esquema común que incluya estos componentes.

- **Servidor de Audio y Video(A/V server):** En servidor de audio y video se administra y gestiona el material audiovisual que será difundido por la estación televisiva, mediante herramientas de software específico para esta tarea se puede automatizar el horario y el material destinado a manera de lista de reproducción, así se consigue que una estación televisiva pueda operar las 24 horas del día, los 365 días del año.
- **Audio y Video “en vivo”(Live A/V):** En algunas ocasiones en lugar de que un servidor de audio y video entregue la señal, se tiene la posibilidad de generar este contenido en el mismo instante en que se transmite, es decir la señal que se genera es el resultado de producciones que se realizan en ese instante, por ejemplo, los programas de noticias, programas de entretenimiento, transmisiones satelitales, transmisiones vía micro onda, etc.
- **Switcher:** permite seleccionar, mezclar y manipular diferentes fuentes de señales de audio y/o vídeo, también permite agregar un logotipo de la televisora, entre cosas adicionales como efectos, etc.
- **Encoder:** Codificador. Dispositivo que convierte una señal SDI⁷² a una señal de video compuesto.
- **Multiplexador:** Recibe la señale compuesta y de datos para formar un solo flujo de datos que se envía al modular.

⁷² Serial Digital Interface

- **Modulador:** Prepara el flujo recibido del multiplexador para enviarla al transmisor de acuerdo a la norma de televisión digital establecida.

Figura 4-17: Esquema de una estación televisiva preparada para la Tv Digital



Fuente: GOMES SOARES, Luiz, JUNQUEIRA BARBOSA, Simone, "TV digital interativa no Brasil se faz com Ginga". 2010

A estos componentes es necesario agregar uno adicional, el servidor de aplicaciones, además también es necesario un bloque que genere el flujo de datos a fin de unirlo al resto de componentes, audio y video. En el caso de GamaTv es necesario implementar a fin de completar su esquema para prepararlo para la televisión digital interactiva.

4.6.1.1 Servidor de aplicaciones(*Application Server*)

Este servidor brinda los servicios de almacenamiento de aplicaciones interactivas desarrollados en dos entornos NCL y Java. Este servidor permitirá emitir las aplicaciones (*Datacasting*, difusión de datos) que serán multiplexados en conjunto con el audio y video en una sola trama TS (*Transport Streaming*).

4.6.2 FUNCIONALIDADES DEL SERVIDOR DE APLICACIONES

El servidor de interactividad presenta varias funcionalidades, que se caracterizan por ser modular. Los módulos son requeridos de acuerdo a las necesidades para el Datacasting. A continuación se mencionan las funciones de un servidor de aplicaciones.

- **Servidor SI:** Multiplexación y generación de sistemas de información. Generación de información para tablas(PAT⁷³, PMT⁷⁴, NIT⁷⁵, EIT⁷⁶, SDT⁷⁷, TDT⁷⁸, TOT⁷⁹, BIT⁸⁰), configuración de número canal virtual, configuración del ID de servicio.
- **Servidor EPG (Guía Electrónica de Programas):** Es un servicio organizado donde se encuentra de manera rápida y sencilla todos los canales que ofrece un distribuidor de programas de televisión.

73 Program Association Table. Tabla de asociación de programa.

74 Program Map Table. Tabla de Mapeo del Programa.

75 Network Information Table. Tabla de información de red.

76 Event Information Table. Tabla de información de eventos.

77 Service Description Table. Tabla de descripción de servicio.

78 Time and Date Table. Tabla de fecha y hora.

79 Time Offset Table. Tabla de diferencia de fecha y hora.

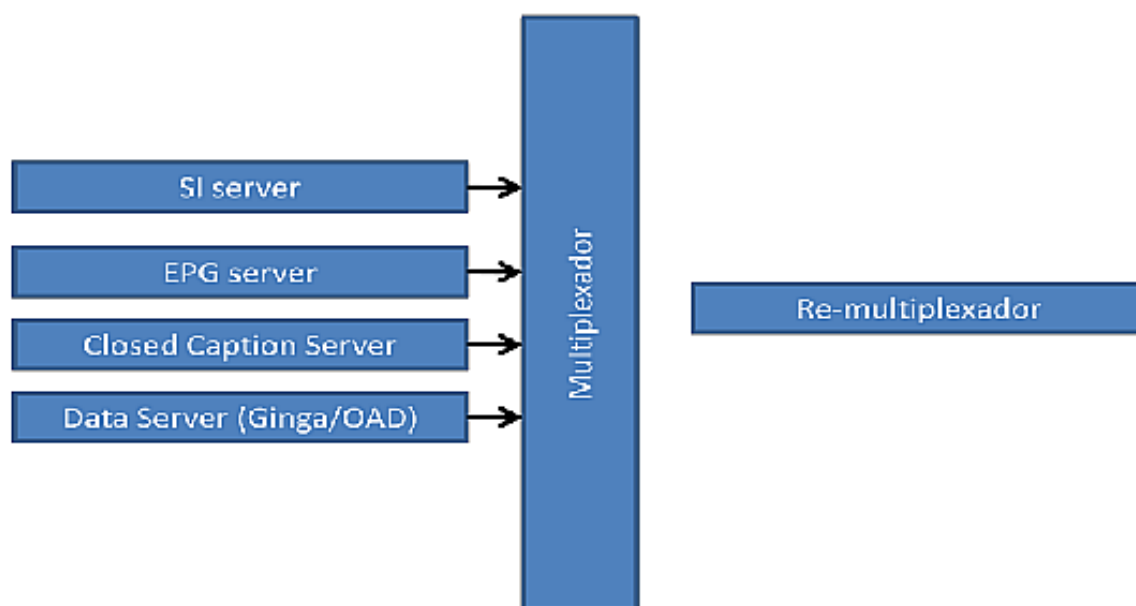
80 Broadcaster Information Table. Tabla de información para radiodifusión.

Durante la multiplexación y generación EPG es donde se definen las tablas para una trama TS⁸¹.

- **Closed caption:** La función de *closed caption* es usado para adicionar subtítulos en flujo transmitido.
- **Object Carousel Server:** Codificación de datos, genera el carrusel de objeto según el protocolo DSM-CC. Soporta aplicaciones en tiempo real de entornos Ginga-J, Ginga-NCL, Ginga-Bridge y GEM.

En la figura 4-18 se muestra los componentes del servidor de aplicaciones.

Figura 4-18: Componentes de servidor de aplicaciones



Fuente: Universidad Nacional de Ingeniería. Instituto Nacional de Investigación y Capacitación de Telecomunicaciones. Lima-Perú. 2010

El multiplexor cumple la función de multiplexar las aplicaciones interactivas almacenadas en el servidor. El video, audio y datos son multiplexados en una trama TS de 188 bytes de tamaño.

El Re-multiplexor cumple la función de multiplexar audio, video y datos en un BTS

⁸¹ Transport Stream permite multiplexar varias señales dentro de un mismo flujo binario de datos.

(*Broadcast Transport Streaming*) de 204 bytes de tamaño. En este caso los flujos de audio y video ingresan al servidor de interactividad a través del puerto ASI (*Asynchronous Serial Interface*), así para el posterior re-multiplexado en formato BTS.

4.6.3 TRANSMISIÓN DE DATOS

La transmisión de datos en un sistema de televisión digital se lleva a cabo a través de un método llamado carrusel de datos, este método consiste en el envío de datos de manera cíclica.

Existen dos mecanismos de envío de datos a través de carrusel que son los siguientes: carrusel de datos y carrusel de objetos.

El mecanismo de carrusel de datos y objetos son los más utilizados para la difusión de datos en los sistemas DVB-T, ATSC-T ISDB-T/Tb. Los dos mecanismos (carrusel de datos y objetos) son protocolos de difusión definidos por el estándar DSM-CC (*Digital Storage Media, Command and Control*).

El protocolo llamado carrusel fue desarrollado como una intuición de posibilidad de difusión de datos, de forma periódica, para set top boxes o televisores digitales. La operación de este protocolo es el envío de módulos de datos de manera cíclica hacia los receptores.

El servidor de aplicaciones puede ser representado en un solo componente de hardware, en la solución proporcionada por la empresa EITV y su producto *Playout Professional*, más información de las características de este equipo en el Anexo H.

4.6.4 BLOQUE DE EDICIÓN Y DESARROLLO

En este bloque generalmente se encuentra la edición y revisión del material audiovisual y donde se debe localizar a los desarrolladores de aplicaciones interactivas, de manera de que los diseñadores gráficos definan la presentación

de las mismas de acuerdo a la “imagen” de GamaTv.

El ambiente de desarrollo que debe ser montado dentro de este bloque puede seguir lo planteado en esta investigación y que se adjunta a este trabajo en el Anexo A.

Este bloque se encuentra integrado con el bloque de pruebas y con el bloque de emisión, concretamente con el servidor de aplicaciones, que enviará las aplicaciones desarrolladas una vez que se haya realizado las pruebas necesarias.

4.6.5 BLOQUE DE PRUEBAS DE APLICACIONES

El bloque de pruebas garantizará que las aplicaciones que se construyan funcionen de la manera correcta, en este bloque se integran todos los componentes del esquema básico de televisión digital: servidor de audio y video, servidor de aplicaciones, encoder, multiplexación, modulación, transmisión y recepción.

Para realizar esta tarea existe en el mercado un producto llamado *Playout* de la empresa EITV, en el Anexo I. se ofrece más detalles de este equipo.

4.7 CONSIDERACIONES FINALES

En este capítulo se remarcó la importancia de las pruebas y como su práctica es aconsejable en todos los sentidos, poder determinar con exactitud que cosas funcionan correctamente y que cosas requieren alguna corrección o mejora hará que los usuarios además de hacerle participe del proceso, reciban un producto con la más alta calidad.

Los ambientes de simulación permiten al desarrollador en todo el proceso de diseño ir verificando que lo que se está programando tiene con finalidad lo que se tiene en mente, estos simuladores ofrecidos en varias alternativas a los programadores dan una aproximación de como una aplicación interactiva se comportaría en un receptor en un ambiente real.

Por último saber de manera más profunda como integrar de la mejor manera el componente nuevo de la televisión digital, la interactividad requiere de equipos que se acoplen a los que ya cuenta en este caso GamaTv a fin de lograr aprovechar todas las características que ofrece el estándar Isdb-t.

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Se pudo determinar que de acuerdo al contenido del material multimedia emitido por la estación televisiva, las aplicaciones interactivas pueden variar en tres grupos definidos: Aplicaciones Interactivas autónomas, donde la aplicación esta deslindándose por completo del contenido emitido; Servicios interactivos asociados a programas, mediante el cual se complementa la información del programa audiovisual con una aplicación interactiva mediante extras o información adicional; y Programas Audiovisuales interactivas, donde el contenido emitido por la estación televisiva está sincronizada con la aplicación interactiva, haciendo uso de recursos como concursos y la interacción directa del usuario, haciendo de la última opción la más compleja de llevar acabo.
- Se pudo observar que las aplicaciones interactivas pueden llegar a ser desde aplicaciones sencillas e independientes, hasta aplicaciones realmente complejas que abarquen un sinnúmero de opciones e incluso participen de la creación del programa audiovisual, también sin dejar de lado la idea de que exista una producción televisiva que se vaya desarrollando de manera que los televidentes interactúan con ella.
- Se comprobó que la capa de middleware permite al desarrollador desentenderse del hardware y centrarse en el desarrollo mismo de la aplicación, haciendo transparente para él las novedades que ha más bajo nivel suelen presentarse, desanimando en muchas ocasiones a programadores recién llegados.
- Se determinó que las aplicaciones interactivas pueden ser construidas en paradigmas diferentes, se puede desarrollar aplicaciones de tipo declarativo, de tipo procedural o ambas, de acuerdo a las necesidades o capacidades de los desarrolladores o de acuerdo al negocio de la empresa televisiva.

- Se procedió a definir como el middleware Ginga y sus implementaciones, NCLua y Java pueden coexistir en un mismo componente, ambos con gran capacidad para dar soporte a cualquiera de las ideas que se tenga en mente dando un gran valor al producto final y haciendo merecedor de una gran confiabilidad a estos lenguajes de programación por su potencia.
- Se comprobó que la opción de compartir el conocimiento es la mejor manera de seguir desarrollando e innovando, en este caso el middleware Ginga, ya que, gracias a que es software libre las instituciones científicas, empresas privadas o universidades, pueden realizar sus trabajos de investigación, a fin de poder aportar mejoras o corrección de errores en beneficio de todos los involucrados en el tema.
- La inclusión en Ginga de la implementación Java(Ginga-J) posibilita que estas aplicaciones interactivas desarrolladas bajo este lenguaje de programación puedan ser intercambiadas entre otros estándares de televisión digital que soporten la norma GEM, abriendo muchas más posibilidades de distribuir nuestras aplicaciones y también de hacer uso de aplicaciones que tienen otra connotación por ser originarias de otro continente.
- Se estableció como la integración perfecta entre lenguajes declarativos en este caso NCL y lenguajes procedurales en este caso Lua permiten realizar aplicaciones tan simples como selección de opciones hasta aplicaciones muy complejas que involucren el uso del canal de retorno con la integración a sistemas mediante el consumo de *web service* o la consulta o servicios específicos en la red.
- Aprender el lenguaje Lua permitirá extender la funcionalidad de nuestras aplicaciones interactivas, Lua permite dibujar en pantalla de manera más rápida y eficiente que NCL, por ejemplo, cada vez que con NCL definimos una región y la mandamos a mostrarse(*start*) el intérprete del middleware refresca la pantalla, si a esta región le sumamos cientos de regiones más, el refresco de pantalla por cada región hará que la respuesta de la acción requiera más

tiempo y más recursos. Si en cambio con Lua definimos una o cientos de zonas(*canvas*) y al final realizamos un solo refresco(*flush*) definitivamente esta acción requerirá menos tiempo y menos recursos.

- La aplicación puede definir dos regiones principales, la región total de la pantalla y la región donde la aplicación Lua se ejecutará, de manera que el archivo principal(*main.ncl* por ejemplo) se definirá como un “lanzador” de aplicaciones Lua, optimizando de la mejor manera a la hora de programar este tipo de aplicaciones.
- Se pudo verificar como la interactividad agrega un valor añadido a la televisión terrestre gratuita, la misma que accede gran parte de la población y que hasta el momento en la era analógica es restrictiva a ciertos usos que en un futuro estarán disponibles para todos.
- La cadena de valor de la era analógica ha pasado de tener: Producción de contenido, Distribución y Consumo; ha tener: Proveedores de Middleware. Proveedor de Aplicaciones Interactivas, Distribución, Consumo y Canal de retorno; abriendo nuevos campos de crecimiento económico, proporcionado fuentes de trabajo y haciendo partícipes a mucha más gente en el proceso de la televisión.

5.2 RECOMENDACIONES

- La herramienta Composer permite a personas interesadas en aprender el lenguaje de programación NCL iniciarse en este tema, se recomienda para programadores novatos o que quieren empezar a programar aplicaciones interactivas, ya que, su curva de aprendizaje es relativamente sencilla además de contar con diferentes opciones de desarrollo que harán en un principio que la experiencia sea positiva.
- El uso de componentes multimedia como imágenes por ejemplo en el desarrollo de aplicaciones interactivas, debe ir cuidadosamente escogida de

acuerdo a lo que la norma exige con la finalidad de evitar que estos archivos sean descartados por el intérprete del middleware por no estar de acuerdo a lo que las especificaciones indican.

- Es recomendable que una vez dominada la herramienta Composer, se migre hacia una herramienta más completa y versátil, en este caso Eclipse y su plugin NCL que hará que el desarrollo se complique en algo, pero sin duda la recompensa de acceder a un entorno de desarrollo verdadero mitigará el esfuerzo que este aprendizaje requiere. Además que desde Eclipse unificaremos el desarrollo, ya que, desde este IDE se puede programar en Lua, NCL y Java.
- Las comunidades de Ginga en los diferentes países donde la norma nipona-brasileña haya sido adoptada, tienen que seguir el camino de la integración a fin de hacer que el futuro de las aplicaciones interactivas dentro de la televisión digital terrestre sea prometedor y no solo una novedad de una nueva tecnología, para ello las comunidades deben llamar la atención de muchos desarrolladores, para que aporten con aplicaciones novedosas y que puedan ser acogidas por empresas televisivas.
- El gobierno de nuestro país debe apropiarse de la tecnología en esta oportunidad que se ha presentado con la televisión digital, promover el desarrollo de esta tecnología con apoyo a los centros de investigación, universidades, comunidad de programadores, para el beneficio de las personas en general, con iniciativas que procuren la aparición de nuevo mercado económico a nivel local, por ejemplo con el nacimiento de empresas nacionales que se involucren con la fabricación de los receptores de televisión digital, para distribuirlo a la población a un costo mucho menor en comparación con receptores adquiridos a empresas multinacionales.
- A todos quienes estén interesados en involucrarse en el desarrollo no solo de aplicaciones interactivas, sino en el desarrollo del middleware Ginga, no solo en el ámbito de la programación, sino también en etapas importantes como son la

documentación, pruebas de funcionamiento detección de errores, etc., esta es la oportunidad que hay que aprovechar, es la ventaja de estar tan lejos pero al mismo tiempo tan cerca unidos por los lazos tecnológicos, a través de las comunidades locales en cada país, en cada región, a aportar para hacer de Ginga un orgullo latinoamericano, cuidando y manteniendo la ideología con la que nació, Ginga es y será software libre.

REFERENCIAS BIBLIOGRÁFICAS

NORMALIZACIÓN

- ABNT NBR 15606-2. Asociación Brasileira de Normas Técnicas “Televisión Digital Terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 2: Ginga-NCL para receptores fijos y móviles – Lenguaje de aplicación XML para codificación de aplicaciones” Sistema Brasileiro de TV Digital Terrestre.
http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNBR15606-2_2007Vc_2008.pdf, 2008.
- ABNT NBR 15606-5. Asociación Brasileira de Normas Técnicas “Televisión Digital Terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 5: Ginga-NCL para receptores portátiles – Lenguaje de aplicación XML para codificación de aplicaciones” Sistema Brasileiro de TV Digital Terrestre,
http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNBR15606-5_2008Ed1.pdf, 2008.
- ABNT NBR 15606-4 Asociación Brasileira de Normas Técnicas “Televisión digital terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 4: Ginga-J – Ambiente para ejecución de aplicaciones de procedimiento” Sistema Brasileiro de TV Digital Terrestre,
http://www.dtv.org.br/download/pt-br/ABNTNBR15606-4_2010Ed1.pdf, 2010.

DOCUMENTOS Y LIBROS

- GOMES, Luiz, FERREIRA Rogério, FERREIRA, Márcio, “Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System ”, Departamento de Informática, Pontificia Universidad Católica de Rio de Janeiro,
ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/journalpapers/2007_04_jbcs_lfgs.pdf,
revisado en enero 2011.

- LEMOS DE SOUZA, Guido, CUNHA, Luiz, COELHO Carlos, “Ginga-J: The Procedural Middleware for the Brazilian Digital TV System ”, Departamento de Informática, Universidad Federal de Paraíba, http://www.tvdi.inf.br/upload/artigos/ginga-j_the_procedural_middleware_for_brazilian_digital_tv_system.pdf, revisado en febrero 2011.
- RIBEIRO, Jean, “Middleware Ginga”, Departamento de Ingeniería, Universidad Federal Fluminense, <http://www.midiacom.uff.br/~debora/fsmm/trab-2008-2/middleware.pdf>, revisado en enero 2011.
- GOMES SOARES, Luiz, JUNQUEIRA BARBOSA, Simone, “TV digital interativa no Brasil se faz com Ginga ”, Departamento de Informática, Pontificia Universidad Católica de Rio de Janeiro, http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_07_lfgs.pdf, revisado en diciembre de 2010.
- SOARES, Carlos, GOMES, Luiz, FERREIRA, Rogerio, JUNQUERIA, Simone, “Construindo Programas Audiovisuais Interativos Utilizando a NCL 3.0 e a Ferramenta Composer”, Laboratorio de Sistemas Multimedia TeleMídia, Pontificia Universidad Católica de Rio de Janeiro, <http://www.ncl.org.br/documentos/TutorialNCL3.0-2ed.pdf>, revisado en diciembre 2010.
- SANT'ANNA Francisco y otros, “Desenvolvimento de Aplicações Declarativas para TV Digital no Middleware Ginga com Objetos Imperativos Lua”, <http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/MCNCLua.pdf>, revisado en marzo 2011.
- GOMES SOARES, Luiz, JUNQUEIRA BARBOSA, Simone, “Programando en NCL 3.0”, Editorial Elsevier, Tercera Edición, 2009.

- CARVALHO, Rafael, FERREIRA, Joel, RIBEIRO, Jean, VARANDA, Julia, MUCHALUAT, Debora, “Introducción a los Lenguajes NCL y Lua: Desarrollando Aplicaciones Interactivas para TV Digital”, Laboratorio MídiaCom, Universidad Federal Fluminense, <http://www.midiacom.uff.br/gtvd/files/apostila.pdf>, revisado en marzo de 2011.
- IERUSALIMSKY, Roberto, “Programming in Lua”, Second Edition. <http://Lua.Org>, 2006.
- Instituto Nacional de Investigación y Capacitación de Telecomunicaciones, Investigación del Estudio del Middleware GINGA, http://www.ginga.org.pe:8080/ginga/doc_template/pdf/Informe_Ginga_2010_AAT.pdf, revisado en abril 2011.
- HIERA, Lucas. Análise de Middlewares no contexto de Sistemas Terrestres de Televisão Digital, <http://www2.dc.uel.br/nourau/document/?view=715>, revisado en abril 2011.

PAGINAS WEB OFICIALES Y COMUNIDADES

- Página Oficial de Ginga Argentina: <http://www.ginga.org.ar/>, revisando en abril 2011.
- Página Oficial de Ginga Brasil: <http://www.ginga.org.br/>, revisando en mayo 2011.
- Página Oficial de Ginga Perú: <http://www.ginga.org.pe/>, revisando en abril 2011.
- Página Oficial de Software Público de Brasil: <http://www.softwarepublico.gov.br/>, revisado en abril 2011.
- Página Oficial de desarrolladores del Middleware Ginga: <http://svn.softwarepublico.gov.br/trac/ginga/>, revisado en mayo 2011.

- Página Oficial del Club NCL de Brasil: <http://clube.ncl.org.br/>, revisado en mayo 2011.
- Página Oficial del Laboratorio TeleMídia <http://www.telemidia.puc-rio.br>, revisado en marzo 2011.
- Página Oficial del Laboratorio LAVID: <http://www.lavid.ufpb.br/>, revisado en marzo 2011.
- Página Oficial de NCL Brasil: <http://www.ncl.org.br>, revisado en febrero 2011.
- Página Oficial de OpenGinga: <http://gingacdn.lavid.ufpb.br/projects/ginga-j>, revisado en febrero 2011
- Página Oficial de la Comunidad Ginga Argentina: <http://comunidad.ginga.org.ar/>, 2011.

TESIS Y MONOGRAFÍAS

- SOUZA JÚNIOR, Paulo José, "LuaComp: Ferramenta de Autoria de Aplicações para TV Digital", Universidade de Brasília - UNB Faculdade de Tecnologia - FT Departamento de Engenharia Elétrica, 2009.
- ALBUQUERQUE, Antônio; LOPES, João, "A Tv Digital no Brasil e o desenvolvimento de aplicações interativas para o middleware Ginga", Universidade Federal de Sergip. 2008.
- BELTRÃO, Mauro. Gingaway, "Uma ferramenta para criação de aplicações ginga-ncl interativas para tv digital", Universidade Federal de Pernambuco, 2008.
- NEVES DE OLIVEIRA, "Tvision. Ferramenta Gráfica para Desenvolvimento de Aplicações Para TV Digital no Formato GINGA-NCL", Universidade Federal de Pernambuco, 2009.

- PUHLMANN, Christian. "Sistema Brasileiro de TV Digital", Universidade Católica de Pelotas, 2008.
- DAMASCENO, Jean Ribeiro, "Middleware Ginga", Universidade Federal Fluminense, 2009.

BLOGS Y PORTAL DE NOTICIAS

- Grupo Ginga Goiás: <http://grupogingagoias.com.br/blog/>, revisando en abril 2011.
- Manoel Campos: <http://manoelcampos.com/tvd/>, revisado en junio 2011.
- Ramiro Nahuel Pol: <http://www.ramiropol.com.ar/category/tv-didital/>, revisado en mayo 2011.
- GINGA-DF: <http://www.gingadf.com/blogGinga/>, 2011 revisado en abril 2011.
- Rafael Carvalho: <http://rafaelcarvalho.tv/>, revisado en junio 2011.

Anexo A. Entorno de Desarrollo

El entorno de desarrollo es el conjunto de herramientas destinadas a ayudar tanto en la codificación como en la ejecución y visualización de nuestras aplicaciones interactivas.

Para la codificación de nuestros aplicativos se usará el entorno de desarrollo IDE Eclipse, el uso de un IDE siempre trae consigo la facilidad de poder programar con mayor rapidez gracias a sus herramientas integradas que nos ayudan en esta labor.

Instalación de Eclipse

Se usará Eclipse Helios que es la versión más actual al momento, de descarga gratuita de su página web⁸². Eclipse está programado en Java por lo que no es necesario instalar nada, basta solo con ejecutar el archivo binario Eclipse que se encuentra dentro de la carpeta que descargó.

Instalando plugin NCL

Se ha desarrollado un plugin para el IDE Eclipse que agiliza en gran medida el desarrollo de aplicaciones con NCL, Lua y Java, a través del módulo de instalación automática de Eclipse es posible hacer sin ningún inconveniente. Para realizar la instalación del plugin procedemos de la siguiente manera.

Accedemos a Help, Install New Software como se muestra en la figura A-1.

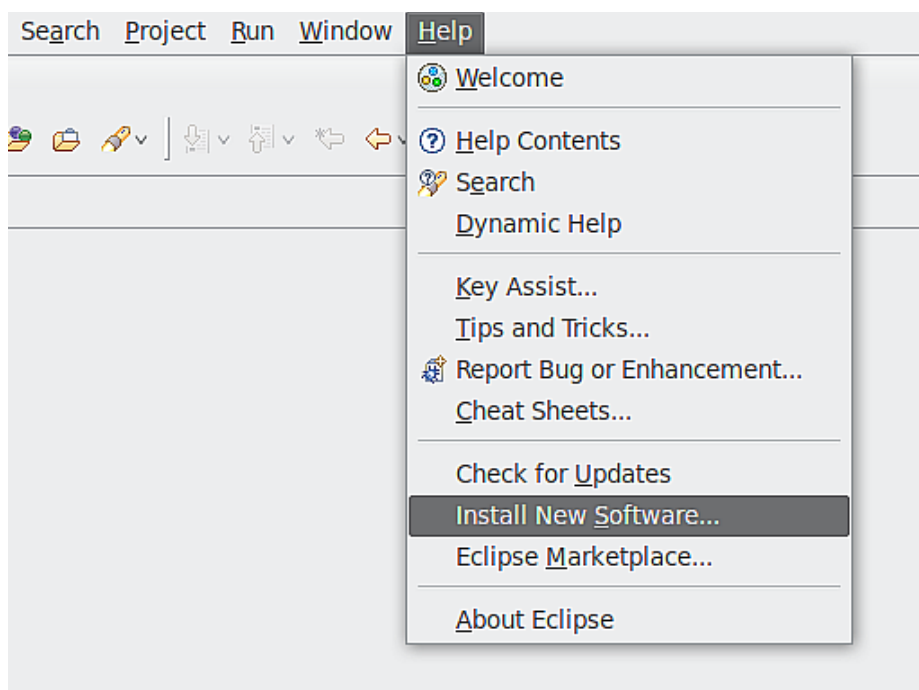
Damos clic en el botón *Add* y se muestra una ventana de diálogo para registrar la información que se solicita del lugar de donde proviene el plugin. Tenemos que ingresar los siguientes datos:

Name: Ginga NCL(nombre descriptivo, puede ser cualquiera)

Location: <http://www.laws.deinf.ufma.br/ncleclipse/update>

⁸² <http://www.eclipse.org/downloads>

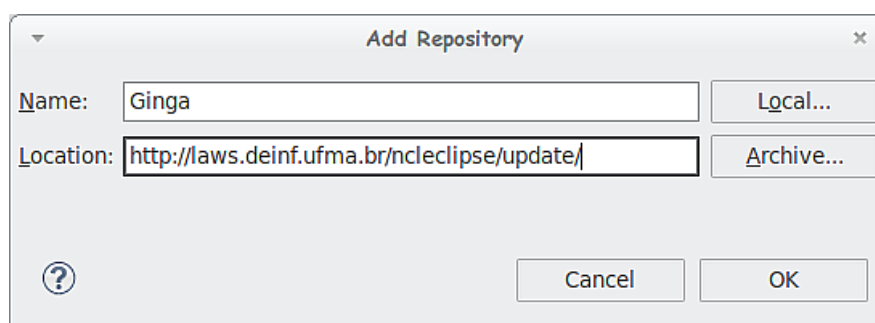
Figura A-1: Instalando Plugin NCL en Eclipse



Fuente: El autor

Como se muestra en la figura A-2 completamos la información para comenzar la descarga del plugin y que estos datos se almacenen para cuando se necesite actualizar este componente.

Figura A-2: Completando la información sobre el plugin NCL



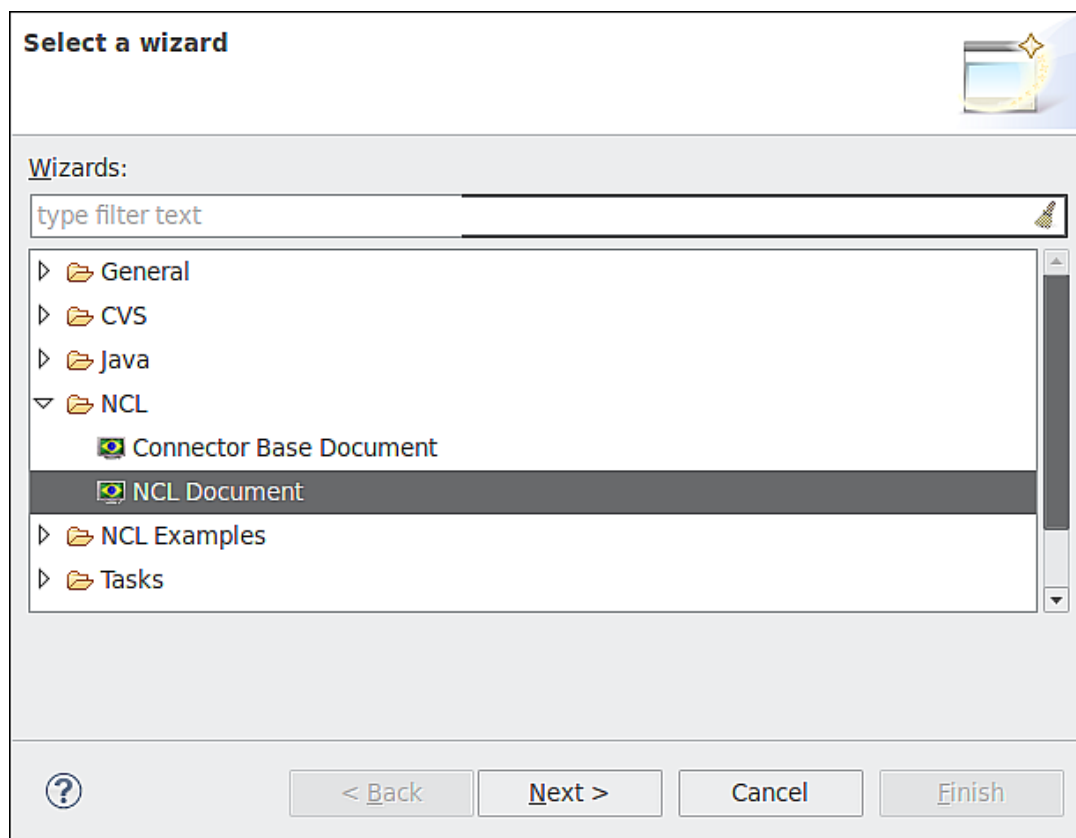
Fuente: El autor

Cuando el plugin aparezca en la lista, marque la respectiva elección, clic en *Next* y posteriormente *Finish* para concluir con el proceso.

Eclipse necesitar reiniciarse para guardar los cambios, luego de esto el IDE está listo para nuestro primer documento NCL.

Para crear un nuevo documento NCL:

Figura A-3: Creando un nuevo documento NCL



Fuente: El autor

File, *New*, *Other* o con el atajo de teclado *Ctrl + N*. En la ventana de dialogo escoger *NCL*, *NCL Document* y clic en *Next*.

Instalación del plugin LuaEclipse

Con el plugin LuaEclipse se puede editar scripts Lua con syntax highlight⁸³, completación de código, verificación de errores de compilación agrupación de código y comentarios, ejecución de scripts utilizando un intérprete pre configurado, etc.

La instalación es la misma que la de cualquier plugin, por lo que es igual a la

⁸³ Es una característica importante de los IDE's para resaltar con colores el código fuente que ayuda a diferenciar las palabras reservadas, variables, comentarios, etc.

instalación del plugin NCL, solo debemos usar la información correcta cuando agregamos el repositorio del plugin con estos datos:

Name: Lua Eclipse

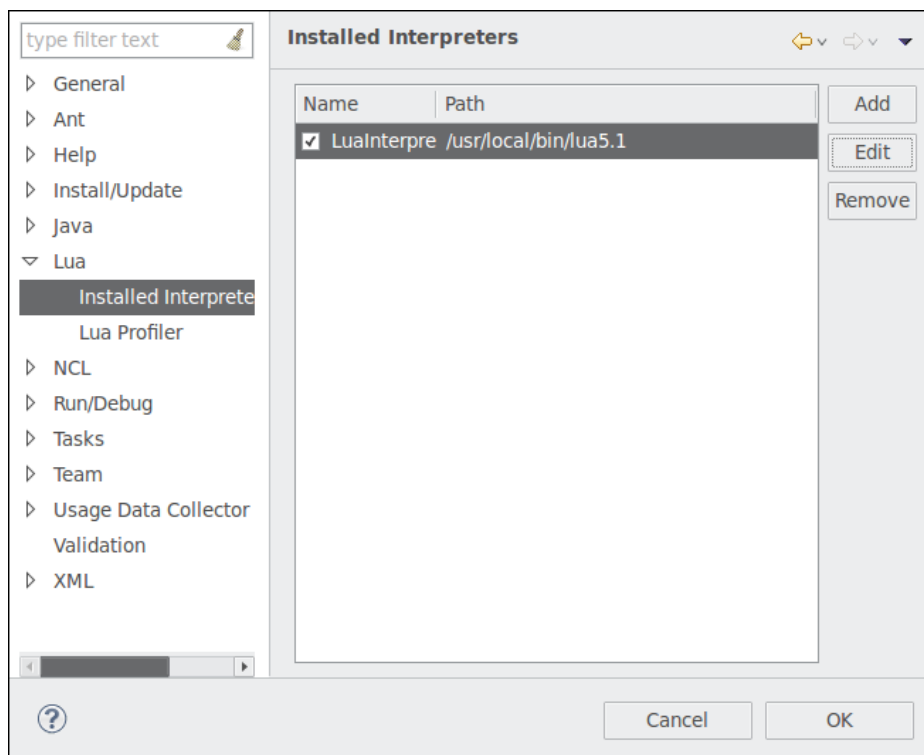
Location: <http://luaeclipse.luaforge.net/preview/update-site/linux.gtk.x86>

Si su sistema operativo es Microsoft Windows, sustituir el valor de:

Location: <http://luaeclipse.luaforge.net/preview/update-site/win32.win32.x86>.

Luego de la instalación necesitamos configurar un intérprete Lua para nuestras aplicaciones. Para ello seleccionamos *Window, Preferences, Lua, Installed Interpreters* como en la figura A-4.

Figura A-4: Configurando un intérprete Lua



Fuente: El autor

Damos clic en Add y completamos la información de la ubicación del intérprete de Lua, generalmente en sistemas Gnu/Linux este intérprete se encuentra en esta

ruta: /usr/local/bin/lua5.1.

En sistemas Windows la ruta del intérprete es posible encontrarla en: Archivos de Programa/Lua/lua5.1.

Guardamos los cambios, si todo ha salido bien ya tenemos instalado el ambiente para programar con Lua.

Para crear un nuevo proyecto Lua damos clic en : *File, New, New Lua Project*. En la ventana de diálogo, complete la información del proyecto y continúe hasta crearlo, ahora para crear el archivo Lua damos clic derecho sobre el proyecto que acabamos de crear y escogemos *New, New Lua File*.

Para ejecutar su programa Lua, tenemos que dar clic derecho sobre el archivo y escoja *Run As, Lua Application*.

Anexo B. Lenguaje NCL

Estructura de un Documento NCL

Un documento NCL es un archivo escrito en XML. Todo documento NCL posee la siguiente estructura:

- Un encabezado de archivo NCL (líneas 1 y 2);
- Una sección de encabezado de programa (sección *head*, líneas 3 – 13), donde se definen las regiones, los descriptores, los conectores y las reglas utilizadas por el programa;
- Un cuerpo de programa (sección *body*, líneas 14-17), donde se definen los contextos, nodos multimedia, enlaces y otros elementos que define el contenido y la estructura del programa;
- Por lo menos una puerta que indica por donde el programa comenzará a ser exhibido (*port pInicio*, línea 15); y
- La conclusión del documento (línea 18).

La figura B-1 presenta la estructura básica de un documento NCL.

Generalmente, los pasos para construir un documento NCL deben definir:

- Los encabezados básicos del archivo NCL y del Programa;
- Las regiones de la pantalla en donde se presentarán los elementos visuales (región Base);
- Como y donde los nodos multimedia serán presentados, a través de descriptores (*descriptorBase*);

Figura B-1: Estructura básica de un documento NCL

Encabezado del Archivo NCL		1: <?xml version="1.0" encoding="ISO-8859-1"?>		1														
Encabezado del Programa		2: <ncl id="exemplo01" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd">																
		3: <head>																
	base de regiones	4: <regionBase>	5: <!-- regiones de la pantalla en donde se presentaran los archivos multimedia -->	6: </regionBase>	7: <descriptorBase>	8: <!-- descriptores que define como los archivos multimedia serán presentados -->	9: </descriptorBase>	10: <connectorBase>	11: <!-- conectores que define como los enlaces son activado y las acciones que ellos ejecutan -->	12: </connectorBase>	13: </head>	14: <body>	15: <port id="plnicio" component="ncPrincipal" interface="Inicio"/>	16: <!-- contextos, nodos multimedia y sus anclas, enlaces y otros elementos -->	17: </body>	18: </ncl>	2	
	base de descriptores	7: <descriptorBase>																8: <!-- descriptores que define como los archivos multimedia serán presentados -->
																		3
																		8
																		5
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
																		7
																		4
																		6
														</				

- El contenido (nodos multimedia - media) y la estructura (contextos - *contex*) del documento (sección *body*), asociados a los descriptores;
- La puerta de entrada al programa, apuntando al primer nodo que va a ser presentado, así como las puertas para los contextos, con el propósito de desarrollar enlaces entre contextos y nodos multimedia (*port*);
- Anclas para los nodos multimedia, con el propósito de construir los enlaces entre nodos multimedia (*area* y *attributte*);
- Enlaces para el sincronismo e interactividad entre los nodos multimedia y contextos (*link*); y,
- Los conectores que especifica el comportamiento de los enlaces del documento (*connectorBase*).

Regiones

Una región (*region*) se define como un área en el dispositivo de salida, donde un nodo multimedia puede ser presentado. Para organizar el diseño de las distintas partes del documento hipermedia, las regiones pueden estar anidadas. En NCL, las regiones son definidas en el encabezado de programa (*<head>*), en la sección de las regiones base (*<regionBase>*).

Todo documento NCL posee por lo menos una región, que define la dimensión y las características del dispositivo donde uno o más nodos multimedia serán presentados. Una región sirve para analizar la posición de los nodos multimedia en un lugar específico.

Por ejemplo, si creamos dos regiones, a la primera la denominaremos rgTV, en la que definiremos las dimensiones de la pantalla de la TV, y la segunda rgVideo1, que servirá para presentar un video en un lugar específico de la TV (asumiendo que las dimensiones del televisor son 1024x576 pixeles y vamos a reproducir un video de 640x480 pixeles de dimensión):

```
<region id="rgTV" width="1024" height="576">
<region id="rgVideo1" left="192" top="48" width="640" height="480"
/>
</region>
```

En NCL, para una región se definen los siguientes atributos:

id*: es el identificador único, utilizado para referenciar una región (por ejemplo, los descriptores de los archivos multimedia que serán representados en la región).

title (título): Es el título de una región. En caso de ser una región exhibida como una moldura, este título será el que aparece como título de la ventana correspondiente.

left (izquierda): Hace referencia a la coordenada “x” del lado izquierdo de la región, con relación a la coordenada del lado izquierdo de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).

top (tope): Hace referencia a la coordenada “y” del lado superior de la región, con relación a la coordenada del lado superior de la región (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).

right (derecha): Hace referencia a la coordenada “x” del lado derecho de la región, con relación al lado derecho de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).

bottom (base): Hace referencia a la coordenada “y” del lado inferior de la región, con relación al lado inferior de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).

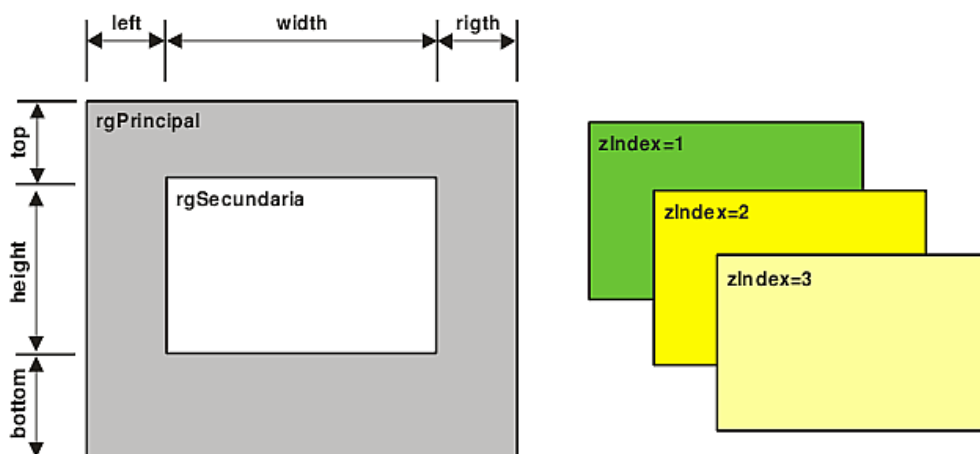
width (anchura) y height (altura): Son las dimensiones horizontal y vertical de una región. Cabe observar que como autor se puede especificar las dimensiones de una región según su conveniencia. Por ejemplo, en ciertos casos puede ser mejor definir los atributos right, bottom, width y height. En otros casos, puede ser

más apropiado especificar los atributos top, left, width y height

zIndex: Indica la posición de una región en el eje “z” y es utilizado generalmente para indicar, en el caso de existir regiones sobrepuestas, que región se presenta sobre las otras. Las capas con índice mayor serán presentadas sobre las capas de índice menor.

En la Figura B-2 se ilustra los atributos top, left, right, bottom, width, height e zIndex:

Figura B-2: Atributos de posición y tamaño de una región



Es importante señalar que si todos los atributos de posición y el tamaño se especifican, los atributos left (izquierda) y width (ancho) tienen prioridad sobre el atributo right (derecho), así como los atributos top (superior) y height (altura) tienen prioridad sobre el atributo bottom (fondo).

Por otra parte, cuando dos o más regiones tienen el mismo valor zIndex, y en cada región que se presenta un archivo multimedia, existen dos posibilidades: el caso que un archivo multimedia sea presentado antes que otro, un archivo multimedia que se ha iniciado después se va a sobreponer a la que fue iniciada anteriormente (orden temporal). En caso de que los dos archivos multimedia sean iniciados al mismo tiempo, el orden será seleccionado aleatoriamente por el programa intérprete.

La base de regiones posee los siguientes atributos:

id: identificador único de la base de regiones.

device: Define el dispositivo al cual la región está asociada. Puede ser de la forma `systemScreen(i)`, `systemAudio(i)`, de acuerdo a los dispositivos disponibles.

Descriptores

Un descriptor es el elemento encargado de definir como será presentado un nodo multimedia, asociándolo a una región (region). En NCL, todos los descriptores deben ser definidos en el encabezado del programa (`<head>`), en la sección llamada base de descriptores (`<descriptorBase>`). Por lo tanto, todo nodo multimedia que será presentado, debe tener asociado un descriptor [20].

En NCL se definen los siguientes atributos para un descriptor:

id*: identificador único, empleado en las referencias de un descriptor (por ejemplo, los archivos multimedia/contenidos presentados por el descriptor).

player: identifica la herramienta de presentación que se empleará para mostrar el objeto multimedia asociado al descriptor. Este atributo es opcional. Cuando se omite, el programa intérprete procura buscar esta herramienta por defecto en función al tipo de objeto multimedia.

explicitDur: determina la duración ideal de un objeto multimedia asociado a un descriptor. El valor de este atributo debe ser expresado en segundos, escribiendo primero el valor numérico seguido de la letra “s” (por ejemplo, 9.9s). Cuando el atributo `explicitDur` no se especifica, el objeto multimedia asociado a este descriptor será presentado con su duración por defecto.

Los archivos multimedia como textos e imágenes estáticas poseen una duración infinita por defecto. Para el programa intérprete, este atributo no se considera para archivos multimedia continuos; en un programa intérprete con soporte de

ajuste elástico del tiempo, el valor de ese atributo es usado como referencia para modificar el tiempo de duración del nodo (permitiendo, por ejemplo, que algunos cuadros de un video de 30s dejen de ser exhibidos para que solo dure 29s).

region: Identifica la región asociada a un descriptor. Al utilizar un descriptor, el objeto multimedia será presentado en su región correspondiente. Este atributo solo precisa ser especificado para objetos que poseen un contenido visual a ser presentado.

freeze: identifica lo que sucede después de la presentación del objeto multimedia asociado a un descriptor. En un video, el valor “true” indica que el último cuadro debe ser congelado.

focusIndex: define un índice de navegación para un objeto multimedia asociado a un descriptor. En el caso de no ser definido un índice, el objeto multimedia no podrá recibir el foco de navegación. En un objeto, el foco inicial estará con el menor focusIndex. Cabe observar que el valor que se le asigna a un focusIndex no es necesariamente un número y en ese caso, “menor” significa lexicográficamente menor.

focusBorderColor: define el color del rectángulo que debe aparecer sobrepuesto en la región de este descriptor cuando el objeto a él asociado recibe el foco. Puede ser uno de los siguientes valores: white, black, silver, gray, red, maroon, fuchsia, purple, lime, green, yellow, olive, blue, navy, aqua, o teal.

focusBorderWidth: define el espesor en pixeles del rectángulo que debe aparecer sobrepuesto a la región de este descriptor cuando el elemento a él asociado recibe un foco. En caso de que este sea igual a 0, ningún borde será presentado. Un valor positivo indica que el borde estará fuera del contenido del objeto, mientras que un valor negativo indica que el borde se formará sobre el contenido del objeto.

focusBorderTransparency: define el porcentaje de transparencia de un borde. Este recibe un valor real entre 0 y 1, donde 0 significa totalmente opaco y 1

significa totalmente transparente.

focusSrc: define un archivo multimedia alternativo que debe ser presentado cuando el elemento asociado a este descriptor esté con el foco.

focusSelSrc: define un archivo multimedia alternativo que debe ser presentado cuando es presionado el botón “Ok” o “Enter” mientras el elemento asociado a este descriptor esté con el foco.

selBorderColor: define un color de borde que debe ser exhibido cuando sea presionado el botón “Ok” o “Enter” mientras el elemento asociado a este descriptor ande con el foco.

moveLeft: identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para la izquierda del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo focusIndex del elemento E).

moveRight: identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para la derecha del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo focusIndex del elemento E)

moveUp: identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para arriba del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo focusIndex del elemento E).

moveDown: identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para abajo del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo focusIndex del elemento E).

transIn: define la transición que será ejecutada al iniciar la presentación del

objeto multimedia.

transOut: define la transición que será ejecutada al terminar la presentación del objeto multimedia.

En NCL se define aún el siguiente elemento opcional contenido en un elemento

descriptor:

descriptorParam: define un parámetro del descriptor como un par *<propiedad,valor>*. Las propiedades y sus respectivos valores dependen del programa de presentación del archivo multimedia asociado al descriptor.

Cada descriptor puede contener diversos elementos *descriptorParam*, definidos en el formato:

```
<descriptorParam name="nombre_del_parametro"
value="valor_del_parametro" />
```

Para indicar que archivo multimedia correspondiente debe ser reproducido con un volumen del 90% del máximo:

```
<descriptor id="dVideo1" region="rgVideo1">
<descriptorParam name="soundLevel" value="0.9" />
</descriptor>
```

El uso de parámetros de un descriptor promueve un alto grado de flexibilidad. Le corresponde a cada programa de presentación de objetos multimedia (player) interpretar esas propiedades de la forma adecuada. Actualmente, no se puede definir parámetros de un descriptor en el Composer. En NCL, están reservados los parámetros que se describen en la Tabla B-1:

Tabla B-1: Parámetros que pueden ser utilizados en descriptor, de acuerdo al archivo multimedia.

Archivos	Parámetros	Descripción
Objetos con Audio	soundLevel, balanceLevel, trebleLevel, bassLevel	Valores entre 0 y 1. En el caso de soundLevel, 0 = mute; 0.5 =volumen a 50%; y 1 = volumen máximo
	location	Posición del objeto multimedia. Se trata de dos números separados por una coma, en el orden <left,top> en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	size	Dimensiones del objeto multimedia. Se trata de dos números separados por una coma, en el orden <width,height> , en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	bounds	Posición y dimensiones del objeto multimedia. Se trata de cuatro números separados por una coma, en el orden <left, top, width, height>, en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	background	Nombres de colores reservados: white, black, silver, gray, red, maroon, fuchsia, purple, lime, green, yellow, olive, blue, navy, aqua, o teal. También puede ser transparent, para el caso de imágenes con transparencia, como algunos GIFs
	visible	true o false
	transparency	Número real entre 0 y 1 indicando transparencia: 0 significa totalmente opaco y 1 significa totalmente transparente
Objetos Visuales	fit	Toma uno de los siguientes valores: fill, hidden, meet, meetBest o slice, donde: - fill = redimensiona el contenido del objeto multimedia para que toque todos los bordes de la región; - hidden: si la altura intrínseca del contenido del archivo multimedia es más pequeña que el atributo height, el objeto necesita ser renderizado a partir del tope y rellenar su altura restante con el color de background; si es mayor, el sobrante debe ser cortado. Idéntico para el ancho y la izquierda. - meet = redimensiona el contenido del objeto multimedia manteniendo sus proporciones hasta alcanzar uno de los bordes de la región. Si haya un espacio vacío a la derecha o en la parte de bajo, debe ser llenado con el color del background. - meetBest = semejante al meet, pero el objeto multimedia no es ampliado en más del doble de las dimensiones originales. - slice = redimensiona el contenido del objeto multimedia manteniendo

		sus proporciones hasta que toda la región sea llenada. Parte del contenido puede ser cortado a la derecha o en la parte inferior del contenido.
	scroll	Toma uno de los siguientes valores: none, horizontal, vertical, both o automatic.
	style	Localización de un archivo de hoja de estilo.

Nodo Multimedia (o Nodo de Contenido)

Un nodo multimedia o de contenido (media node o content node) define el objeto multimedia propiamente dicho: video, audio, texto, etc. Cada definición de un nodo multimedia debe presentar, además del archivo de origen, el descriptor que regulará la presentación de un objeto específico.

Un nodo multimedia posee los siguientes atributos:

Id: identificador único del nodo multimedia, empleado para hacer referencia al nodo (por ejemplo, en las puertas de los nodos de contexto que se dirigen hacia los archivos multimedia)

Type: tipo de archivo multimedia, especificado como MIME, según la tabla B-2:

Tabla B-2: Tipos de archivo multimedia

Valor de type	Extensión de archivo de atributo src
Text/html	.htm, .html
Text/css	.css
Text/xml	.xml
Image/bmp	.bmp
Image/png	.png
Image/gif	.gif
Image/jpeg	.jpg
Audio/basic	.wav
Audio/mp3	.mp3
Audio/mp2	.mp2
Audio/mpeg4	.mp2
Video/mpeg	.mpeg, .mpg
Application/x-ginga-NCLua	lua
Application/x-ginga-NCLet	.xlt, .xlet, .class
Application/x-ginga-settings	No tiene archivo asociado. Se trata de un nodo de atributos globales para ser utilizado en normas y switches.
Application/x-ginga-time	No tiene archivo asociado (no se define el atributo src)

Src: fuente del objeto multimedia. Se trata del camino para el archivo multimedia. Ese camino puede ser relativo, a partir del directorio donde se encuentra el archivo NCL, o absoluto, a través de una URI. Las URI válidas son las que se presentan en la tabla :

Tabla B-3: URI válidas

Esquema	formato	Uso
File:	///camino_archivo/#id_fragmento	Archivos locales
http:	//id_servidor/camino_archivo/# id_fragmento	Archivos remotos bajados del canal de retorno utilizando el protocolo http
Rstp:	//id_servidor/camino_archivo/# id_fragmento	Streams bajados del canal de retorno utilizando el protocolo rstp
Rtp:	//id_servidor/camino_archivo/# id_fragmento	Streams bajados del canal de retorno utilizando el protocolo rtp
Isdtv-ts:	//id_programa	Streams recibidos del transport stream

Descriptor: identificador del descriptor que controla la presentación del objeto multimedia.

Refer: referencia a otro nodo multimedia previamente definido, como forma de reutilización de nodo (utiliza los atributos del nodo multimedia referenciado, excepto el id)

newInstance: establece si un nodo que se refiere a otro genera una nueva instancia del objeto en el programa intérprete o se reutiliza la instancia previamente creada.

Contextos

El elemento body es un caso particular de contexto, representando el documento como un todo. Los Contextos o nodos de composición son utilizados para estructurar un documento hipermedia, los mismos que pueden ser anidados, con el objetivo de reflejar la estructura del documento y ayudarle al autor a organizar de mejor manera los segmentos del programa audiovisual interactivo.

Los atributos de un contexto son:

Id: identificador único de contexto

Refer: referencia a otro contexto previamente definido (utiliza los atributos del contexto referenciado, excepto el id)

El contexto body de un documento NCL hereda el id del propio documento.

Puertas

Una puerta (port) es un punto de interfaz (interface point) de un contexto, que ofrecen acceso externo al contenido (nodos internos) de un contexto. En otras palabras, para que un enlace apunte a un nodo interno al contexto, este contexto debe poseer una puerta que lo dirija hacia el nodo interno deseado.

En todo documento, debe haber por lo menos una puerta de entrada (port en la sección body) indicando cual es el nodo multimedia o contexto inicial.

```
<port id="pInicio" componet="video1" >
```

Cabe observar que la puerta plnicio del body generalmente se asigna al video principal video1. Esto significa que, al iniciar el documento hipermedia, el programa intérprete seguirá la puerta plnicio, la misma que se dirige hacia el nodo de contenido video1, que será entonces presentado.

Una puerta posee los siguientes atributos:

id: identificador único de la puerta, utilizado en las referencias para la misma.

Component: nodo de media o contexto al cual la puerta se asigna.

En caso de que el *component* sea un contexto, se debe definir el atributo *interface*, haciendo la asignación a una puerta o ancla de aquel contexto. En caso de que el *component* sea un nodo multimedia, se puede definir el atributo

interface como una ancla del nodo. Si el atributo *interface* fuera omitido, todo el nodo será considerado como asignado a aquella puerta.

Interface: nombre del punto de interfaz (puerta) de destino en el contexto o nombre de la ancla de destino en el nodo multimedia o contexto.

Base de Conectores

Todos los conectores son definidos en una base de conectores (elemento `<connectorBase>`, que posee como único atributo un identificador (id) de base.

Una base de conectores debe contener los siguientes elementos hijos:

<causalConnector>: define un conector propiamente.

<importBase>: permite importar una base de conectores de algún otro archivo.

Una base de conectores puede ser definida conforme las siguientes estructuras:

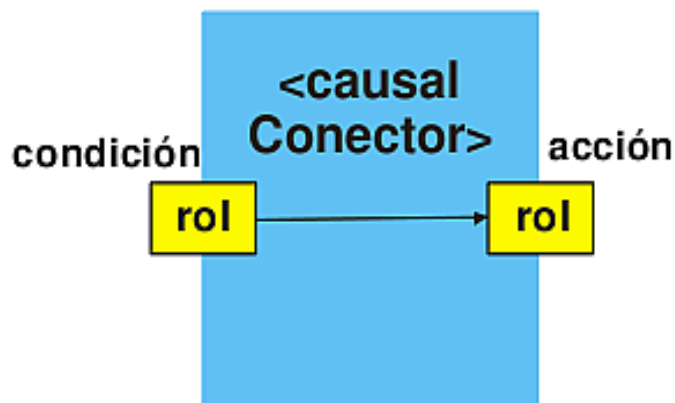
```
<connectorBase id="menusConectores">
  <importBase .../>
  <importBase .../>
  <causalConnector id="onBeginStar">
    ...
  </causalConnector>
  <causalConnector id=" ">
    ...
  </causalConnector id=" ">
    ...
  <causalConnector id=" ">
  </connectorBase>
```

Conectores

En NCM y en NCL, el sincronismo no está hecho por marcas de tiempo (timestamps), pero si por mecanismos de causalidad y restricción definidos en los conectores (connectors). El conector define los papeles (roles) que los nodos de origen y de destino ejercen en los enlaces que utiliza el conector.

En NCL 3.0, existe apenas un tipo de conector: o conector causal (causal Connector). Un conector causal define las condiciones (condition) bajo las cuales el enlace <link> puede ser activado, y las acciones (action) que serán realizadas cuando el mismo sea activado. Un conector causal debe poseer por lo menos condición y una acción. Cada condición u acción está asociada a un papel (role), punto de interfaz que participa de las asignaciones del enlace (Figura B-3).

Figura B-3: Ilustración de un conector causal (elemento <causalConnector>)



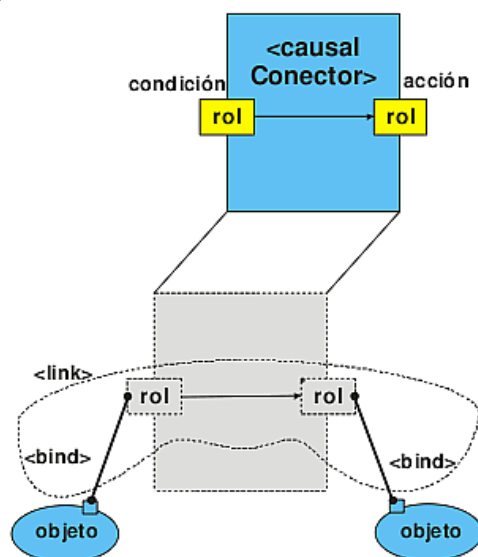
El elemento <link> hace referencia a un <causalConnector> y define un conjunto de asignaciones (elementos <bind> hijos del elemento <link>), que asocian cada extremo del enlace (interface de objeto) a un papel de conector utilizado, como ilustra la figura B-3:

Los elementos hijo de un <causalConnector> son:

<connectorParam>: define parámetros de conector cuyos valores deberían ser definidos por los enlaces que utilizan un conector.

<simpleCondition> y <compoundCondition>: definen las condiciones simples o compuestas de activación de un enlace que utiliza un conector; **<simpleAction> y <compoundAction>**: definen las acciones simples o compuestas que se realizarán cuando un enlace que utiliza un conector sea activado.

Figura B-4: Ilustración de un enlace



Los siguientes papeles de condición son predefinidos (Tabla B-4):

Tabla B-4: Papeles predefinidos de condición

papel	Descripción (el enlace será activado cuando...)
onBegin	...la presentación del nodo multimedia asociado al papel onBegin será iniciada
onEnd	...la presentación del nodo multimedia asociado al papel onEnd será terminada (naturalmente o por un evento stop)
onAbort	...la presentación del nodo multimedia asociado al papel onAbort será abortada
onPause	...la presentación del nodo multimedia asociado al papel onPause será pausada
onResume	...la presentación del nodo multimedia asociado al papel onResume será retomada (después de una pausa)
onSelection	...una tecla <key> será presionada, o cuando la tecla OK sea presionada mientras un nodo multimedia este corriendo.
onAttribution	...un valor <valor> será atribuido

Los siguientes papeles de acción son predefinidos (Tabla B-5):

Tabla B-5: Papeles predefinidos de acción

papel	Descripción (cuando el enlace sea activado...)
Start	...inicia la presentación del nodo multimedia asociado al papel start
Stop	...termina la presentación del nodo multimedia asociado al papel stop
Abort	...aborta la presentación del nodo multimedia asociado al papel abort
Pause	...pausa la presentación del nodo multimedia asociado al papel pause
Resume	...retoma la presentación del nodo multimedia asociado al papel resume (en caso de que este en pausa)
Set	...establece un valor <valor> para el ancla asociada al papel set

Tanto los papeles de condición como los de acción están asociados a transiciones de estados en una máquina de eventos.

Tanto los papeles de condición “onBegin”, “onEnd”, “onAbort”, “onPause” y “onResume”, así como los papeles de acción “start”, “stop”, “abort”, “pause” y “resume” están relacionados a posibles transiciones de estados de presentación de anclas de contenidos.

Un papel de condición “onSelection” está relacionado a eventos de selección y ligado a interactividad a través de dispositivos de entrada, como el control remoto de la TV.

Enlaces

Los enlaces (links) asocian nodos a través de conectores (connectors), que definen la semántica de la asociación entre los nodos. NCL define los siguientes atributos:

Id: identificador único de enlace

xconnector: identificador de conector asociado al enlace

NCL define los siguientes elementos contenidos en un elemento de tipo enlace:

LinkParam: define un parámetro del enlace como un par <propiedad, valor>. Las propiedades y sus respectivos valores dependen de la definición del conector al

cual el enlace está asociado. Un enlace puede contener diversos elementos `linkParam`.

Bind: indica un componente (nodo multimedia o de contexto) involucrado en el enlace, indicando su papel (role) en el mismo, conforme la semántica del conector.

En algunos casos se debe indicar también el punto de interfaz (interface) del nodo, al cual el enlace está ligado (puerta del contexto o ancla de un nodo multimedia). Un enlace puede contener diversos elementos `bind`, y debe contener por lo menos un `bind` para cada papel definido en el conector.

El elemento `bind` puede contener una o más instancias del siguiente elemento como elementos hijos:

bindParam: define un parámetro específico del `bind` como un par <propiedad, valor>. Las propiedades y sus respectivos valores dependen de la definición del conector al cual el enlace está asociado.

A continuación se muestra el esquema básico de conformación de un enlace (<link>):

```
<link id="nombre_de_link" xconnector="id_de_Link">
<bind role="id_del_papel_de_condicion"
      component="id_de_un_objeto"
      interface="id_de_una_interface"/>
  <bind role="id_del_papel_de_accion"
        component="id_de_un_objeto"
        interface="id_de_una_interface"/>
</link>
```

Anclas

Las anclas son puntos de entrada para los nodos multimedia o contextos. El objetivo de utilizar anclas es emplear segmentos o propiedades de un nodo multimedia o contexto, sea como origen o destino de los enlaces. Una ancla

puede ser del tipo ancla de contenido (content anchor) o ancla de propiedad (property anchor).

Un ancla de contenido define un segmento multimedia (intervalo de tiempo y/o región en el espacio) que pudiera ser utilizado como punto de activación de los enlaces. Un segmento de multimedia es una selección contigua de unidades de información (information units) de un nodo. La definición de esas unidades de información depende del tipo de archivo multimedia presentado por el nodo. Las unidades de información de un video, por ejemplo, pueden ser los frames de video.

Un ancla de contenido es definida como un elemento *area* dentro del elemento *media*.

En el siguiente ejemplo, son definidas tres anclas de contenido para un nodo de video:

```
<media type="video" id="video1" src="media/video1.mpg"
descriptor="dVideo1">
<area id="a01" begin="5s" end="9s"/>
<area id="a02" begin="10s" end="14s"/>
<area id="a03" begin="15s" end="19s"/>
</media>
```

NCL define los siguientes atributos de ancla de contenido:

id: identificador único de ancla

coords: coordenadas en pixeles del ancla espacial (atributo válido para archivos multimedia visuales), en el formato "X,Y,width,height"

begin: inicio del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)

end: termino del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)

dur: duración del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)

first: cuadro/muestra del archivo multimedia definiendo el inicio del ancla (atributo válido para archivos multimedia continuos)

last: cuadro/muestra del archivo multimedia definiendo el fin del ancla (atributo válido para archivos multimedia continuos)

text: texto del ancla en el archivo de origen (atributo válido para archivos multimedia de texto)

position: posición del texto del ancla en el archivo de origen (atributo válido para archivos multimedia de texto)

anchorLabel: identificador del ancla en el archivo de origen, tal como es interpretado por la herramienta de exhibición.

Las anclas de propiedad se refieren a propiedades de un nodo de origen o de destino, que pueden ser manipuladas por los enlaces. Ejemplos de propiedades son: volumen del audio de un nodo de audio o video, coordenadas y dimensiones de exhibición de un nodo multimedia visual, entre otros.

Un ancla de propiedad es definida como un elemento property dentro del elemento media o context. En el siguiente ejemplo se definen cuatro anclas de propiedad para un nodo de video, además de un ancla de contenido:

```
<media type=" video" id="video1" src="media/video1.mpg"
descriptor="dVideo1">
<!--anclas de propiedad que seran manipuladas por los enlaces -->
<property name="top"/>
<property name="left"/>
<property name="width"/>
<property name="height"/>
```

```

<!--anclas de contenido en el video que debe sincronizar la imagen
-- >
<area id="aVideo1Image1" begin="3s" end="8s"/>
</media>

```

Reglas

Las reglas simples de un documento NCL son definidas en la sección *ruleBase*, con base a una propiedad, operador y valor, como en el ejemplo a continuación:

```

<ruleBase>
<rule id="rEn" var="idioma" comparator="eq" value="en" />
<rule id="rPt" var="idioma" comparator="eq" value="pt" />
</ruleBase>

```

Los siguientes operadores pueden ser utilizados como comparador en la definición de reglas:

- eq (equal – igual a);
- ne (not equal –diferente de);
- gt (greater than – mayor que);
- ge (greater than or equal to – mayor que o igual a);
- lt (less than – menor que);
- le (less than o equal to – menor que o igual a).

Una forma de almacenar las propiedades que serán utilizadas en las reglas es utilizar el nodo settings.

Anexo C. Lenguaje Lua

Convenios Léxicos

En Lua, los nombres o identificadores pueden ser cualquier cadena de texto, dígitos y guiones bajos, no puede iniciar con un número. Los identificadores se utilizan para nombrar funciones, las variables y campos de una tabla.

Algunas palabras están reservadas y no pueden ser utilizadas como nombres. La tabla muestra la lista de palabras reservadas en el lenguaje.

Tabla C-1: Palabras reservadas del lenguaje Lua

and	break	do	else	elseif	
end	false	for	function	if	
in	Local	Nil	Not	or	
Repeat	return	then	true	while	until

Lua diferencia entre minúsculas y mayúsculas. Se interpretará de manera diferente un fOr o un foR o un FOR o un FOr o un FoR.

Una cadena (de caracteres) se puede definir con comillas simples o dobles, pares y pueden contener secuencias de escape al estilo de C.

Tipos y variables

La Variable es un objeto (una posición, a menudo se encuentra localizada en la memoria), capaz de retener y representan un valor o expresión.

Cuando se hace referencia a la variable, desde el punto de vista de la programación informática, se trata de una región "de la memoria (el equipo), previamente identificados, con propósitos de almacenar los datos o información

de un programa para un determinado espacio de tiempo. "

Tipos

Lua es un lenguaje de tipos dinámicos. Eso no significa que es necesario declarar el tipo de una variable, sino que se crea forma dinámica en función del valor que esta variable ha almacenado. Hay ocho tipos básicos en Lua: nil, boolean, number, string, userdata, function, thread, y table.

Variables

Los nombres de las variables debe comenzar con letra o guión bajo. Para asignar un valor a una variable, se utiliza el operador de asignación. Si hay más valores de las variables, los valores adicionales se descartan. De lo contrario, las variables adicionales son nulas.

Operadores Aritméticos

Lua soporta los operadores aritméticos más utilizados:

- Suma;
- Resta;
- Multiplicación;
- División;
- Exponenciación;
- Negación.

Operadores Lógicos

Los operadores lógicos son:

- and;
- or;
- not.

Funciones

En informática, específicamente en el contexto de la programación, una función (subrutina, procedimiento o subprograma) es una parte de código que resuelve un problema muy específico, que forma parte de un problema mayor (la aplicación final).

Las funciones de Lua son el principal mecanismo de abstracción y de expresión. Se puede realizar tanto una tarea (conocido como un procedimiento o subrutina) para calcular y devolver valores.

Tablas

Tablas se crean mediante el constructor `{}` y se pueden crear de varias maneras:

Para crear una tabla vacía: `tabla = {}`.

Integración NCL-Lua

La importancia de la integración entre las lenguas NCL y Lua se debe principalmente al hecho de que NCL es un lenguaje declarativo, el inconveniente de la falta de un mejor control de la información procesada no se aplica para un lenguaje procedural como Lua, por ejemplo, no es fácil de implementar y permitir la entrada de datos sólo con NCL.

Scripts NCLua

Los string NCLua sirven para utilizar la misma abstracción de objetos multimedia utilizados como imágenes, vídeos y otros medios. En otras palabras, un

documento se refiere únicamente a objetos multimedia NCL, sin importar el tipo de su contenido. El archivo Lua debe estar escrito en un archivo (con extensión. lua) separado del documento NCL.

La principal peculiaridad de NCLua es el hecho de que su ciclo de vida es controlado por un documento NCL. En las funciones se pueden asignar un valor de tiempo como un parámetro que puede ser usado para especificar el momento exacto en que el comando debería ser ejecutado.

La ejecución de un modelo en NCLua es orientada a eventos, es decir, el flujo de la aplicación es manejado por eventos externos del formateador NCL. La activación de estos acontecimientos es controlada por el formateador de NCL. Este comportamiento se caracteriza por un puente de comunicación bidireccional entre el formateador y NCLua.

Módulos NCLua

Las bibliotecas disponibles para NCLua se dividen en cuatro módulos principales que son: módulo event, módulo canvas, módulo settings y el módulo persisten.

MODULO EVENT:

Permite la comunicación del NCLua con el formateador NCL a través de eventos. Para que un NCLua recibe un evento en el formateador, el script debe por lo menos registrar una función de tratamiento de eventos. A partir de entonces cualquier acción tomada por la aplicación en respuesta a un evento recibido.

```
function handler (evt)
    -- codigo para tratar los eventos
end
event.register(handler)    - -registro del formateador
```

Un NCLua también puede enviar eventos para que se comuniquen con el ambiente. Para esto se utiliza la función *event.post*, como se detalla a continuación:


```
event.post (dst; evt)
```

Los eventos son tablas de Lua y el campo *class* es el responsable por identificar la clase del evento. Se puede definir cinco clases distintas:

ncl: Un NCLua que se comunica con el documento en el que se inserta a través de esta clase eventos.

Por ejemplo un NCLua recibiendo un evento, se puede apreciar a continuación:

```
{class='ncl', type='presentation', action='start'}
```

Esto indica que el nodo Lua comenzará su presentación. Y para que un NCLua presente un evento al formateador se procede de la siguiente manera:

```
event.post {class='ncl', type='presentation', action='stop'}
```

Esto indica que el NCLua parará su ejecución.

key: se utiliza para representar el uso del control remoto.

tcp: Uso del canal de retorno realizado por la clase de eventos.

sms: enviar y recibir mensajes de texto.

user: permite crear eventos propios. Como eventos de esta clase son para uso interno.

Para la clase de eventos NCL se utiliza los siguientes campos:

type: puede ser del tipo *presentation*, del tipo *attribution* o del tipo *selection*.

action: puede asumir los siguientes valores: *start*, *stop*, *pause*, *resume* o *abort*.

value: Recibe el valor de la propiedad definida en el campo *property*

A continuación un ejemplo de como se definen estos eventos:

```
event.post {
  class      = 'ncl',
  type       = 'attribution',
  property   = 'myProp',
  action     = 'start',
  value      = '10',
}
```

Para la clase de eventos *key* utilizamos los siguientes campos.

type: debe ser del tipo *press* o del tipo *release*.

key: valor de la tecla en cuestión

Funciones del modulo event

event.register(post, f, class): registra una función enviada al formateador de eventos. El parámetro *post* es la posición de inserción de la función del formateador, indica la posición en la que el parámetro *f* está registrada, si no se especifica este parámetro la función será la última en recibir eventos. El parámetro *f* es la función del formateador. El parámetro *class* es el filtro de la clase, es opcional.

event.unregister (f): cancela el registro de la función enviada al formateador de eventos, donde *f* es la función del formateador.

event.post (dst, evt): genera un evento posterior, donde *dst* es el destino del evento, este parámetro puede asumir valores *in* (envío a la propia aplicación) y *out* (envío al formateador), si no se especifica se asume el valor *out*. El parámetro *evt* es el evento a ser pasado. Retorna *sent* si el evento se envió correctamente o retorna *err_mgs* mensaje en caso de error.

event.timer (time, f): crea un cronómetro, cuando termina su conteo se ejecuta *f*. El parámetro *time* es el tiempo en milisegundos, el parámetro *f* es la función a

ejecutar. Este evento retorna *unreg* que cuando se ejecuta cancela el *timer*.

event.uptimer (time): Devuelve el tiempo en milisegundos recorridos desde el inicio de la aplicación.

Modulo canvas: En NCLua se tiene la posibilidad de hacer operaciones gráficas durante la presentación de una aplicación, tales como líneas, círculos, imágenes, etc. Todo esto es posible gracias a este módulo que ofrece una API gráfica.

Función del módulo canvas

canvas:new(width, height): creación de un nuevo objeto gráfico con un tamaño específico, donde el parámetro *width* establece el ancho del canvas y el parámetro *height* establece el alto del área de trabajo.

canvas:new(image_path): una imagen es pasada como parámetro, donde *image_path* es la ruta de la imagen.

canvas:attrSize(): devuelve las dimensiones del área de dibujo, los parámetros que retornan son *width* y *height* en ese orden, no es posible modificar el ancho y el alto de un área de dibujo ya instanciada.

canvas:attrColor(R, G, B, A): determinar un color con parámetros RGB, el argumento *A* varía de 0(totalmente transparente) a 255(totalmente opaco). Las primitivas son diseñadas en base a su color inicial(0,0,0,255), también es posible pasar el nombre del color *canvas:attrColor(nombre_color, A)*, donde *nombre_color* puede asumir los siguientes valores 'white', 'aqua', 'lime', 'yellow', 'red', 'fuchsia', 'purple', 'maroon', 'blue', 'navy', 'teal', 'green', 'olive', 'silver', 'gray' y 'black'.

canvas:attrClip(x, y, width, height): Limita el área del canvas para el desarrollo. Donde *x* es la coordenada *x* del área limitada, *y* es la coordenada *y* del área limitada, *width* es el ancho del área limitada y *height* es el alto del área limitada.

canvas:attrFont(face, size, style): Altera la fuente del canvas, donde *face* es el

nombre de la fuente, en la norma estas son las fuentes disponibles obligatoriamente *'Tiresias'*, *'Verdana'*. El parámetro *size* es el tamaño en pixeles de la fuente escogida. El parámetro *style* es el estilo para la fuente escogida, entre las cuales están *'bold'*, *'italic'* o *'bold-italic'*.

canvas:attrRotation (degrees): configura el atributo de rotacion del canvas. El parámetro *degrees* rota el área de dibujo en grados.

canvas:attrScale (w, h): escala es el área de dibujo con nuevas dimensiones. El parametro *w* define el ancho del área. El parámetro *h* define la altura del área.

canvas:drawLine (x1, y1, x2, y2): dibuja una línea con referencia en los siguientes puntos(x1,y1) y (x2,y2).

canvas:drawRect (mode, x, y, width, height): dibuja un rectángulo. El parámetro *mode* indica el modo de diseño, *x* es la Coordenada x del rectángulo, El parámetro *y* es la cordenada y del rectángulo, *width* es el ancho del rectángulo, *height* es el alto del rectángulo

canvas:drawRoundRect(mode, x, y, width, height, arcWidth, arcHeight): dibuja un rectángulo con las esquinas redondeadas. El parámetro *arcWidth* es el ancho del arco de la esquina redondeada. *ArcHeight* es el alto del arco de la esquina redondeada, los demás parámetros funcionan de la misma forma que en *canvas:drawRect*.

canvas:drawPolygon(mode): dibuja un polígono. Donde el parámetro *mode* es el modo en el que se va a dibujar el polígono.

canvas:drawEllipse (mode, xc, yc, width, height, ang start, ang end): dibuja una eclipse, donde *mode* es el modo de diseño, el parámetro *xc* es el centro de la elipse , *yc* es el centro de la elipse , *width* es el ancho de la elipse , *height* es el alto de la elipse *ang start* es el ángulo de inicio, *ang end* es el ángulo de fin.

canvas:drawText (x, y, text): dibuja un texto. Donde *x* es la coordenada del texto,

y es la coordenada del texto, *text* es el texto a ser dibujado, dibuja un texto utilizando la fuente configurada en *canvas:attrFont()*.

canvas:clear (x, y, w, h): limpia el canvas con el color configurado en *attrColor*. Donde *x* es la coordenada del área a limpiar, *y* es la coordenada del área a ser limpiada, *w* es el ancho del área a limpiar, *h* es la altura del área a limpiar.

canvas:flush (): actualiza el área de dibujo volviendo a dibujar lo que ya existía en la composición y los nuevos elementos.

canvas:pixel (x, y, R, G, B, A): altera el color de un píxel de un área de dibujo. Donde *x* es la posición del píxel, *y* es la posición del píxel, *R* es el componente de color rojo, *G* es el componente de color verde, *B* es el componente de color azul. *A* es el componente alfa, *canvas:pixel (x, y)* retorna un color de un píxel del área de dibujo.

canvas:measureText (text): devuelve las medidas para el texto pasado. Donde *text* es el texto a ser medido

Devuelve *dx* e *dy*, ancho y alto del texto respectivamente.

Módulo settings: Exporta una tabla de ajustes a las variables definidas por el autor del documento NCL y variables reservadas contenidas en el *application/x-gingasettings*. Por ejemplo, en un objeto NCLua, una variable *settings* como “*system.CPU*” por ejemplo, es mencionada como *settings.system.CPU*.

Módulo persisten: Las aplicaciones NCLua permiten que los datos sean guardados en un área específica del middleware y que se puedan recuperar entre diferentes ejecuciones. Un manejador Lua define un área reservada inaccesible a objetos NCL. No existe ninguna variable predefinida o reservada a objetos procedurales que puedan atribuir valores a esas variables directamente. El uso de una tabla persistente es semejante al uso de una tabla *settings* excepto por el motivo de que los objetos procedurales pueden cambiar los valores de los campos.

Anexo D. Compilación e Instalación de Ginga-NCL

Ginga-NCL versión C++ es una implementación de alto rendimiento, un prototipo ideal para la inclusión en set-top boxes, se caracterizan por bajas capacidades de procesamiento y almacenamiento. La compilación Ginga-NCL tiene requisitos avanzados que no se suele encontrarse en los sistemas de escritorio. Se necesita de cierta experiencia en la compilación del kernel y administración de servicios para seguir con éxito esta guía.

Obtener los paquetes de Ginga-NCL

El código fuente Ginga-NCL está disponible a través del repositorio Ginga en SVN do Portal do Software Público Brasileiro. Los siguientes comandos permiten obtener el código fuente:

```
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/te
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/te
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/gi
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/nc
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/gi
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/gi
svn co --username SuEmail@Provedor.com
http://svn.softwarepublico.gov.br/svn/ginga/gi
```

Sustituya “SuEmail@Provedor.com” por el mail con el cual se registró en la Comunidad Ginga.

Principales requisitos para la compilación e instalación

Cada paquete tiene un conjunto individual de dependencias que deben cumplirse antes de su compilación e instalación. Los pre requisitos enumerados a continuación son consideradas críticos debido a la dificultad de instalación/configuración y las dependencias son comunes en todos los paquetes.

- Kernel de Linux >= 2.6.13 con framebuffer activado y operando.
- DirectFB 1.0.1 funciona correctamente = (DirectFB)
- directfb-examples = 1.0.1 funciona correctamente (DirectFB)
- DirectFB-extra = 1.0.0 (DirectFB) - NOTA: Dentro de la carpeta "interfaces" modificar los archivos *Makefile.am* de forma que todas las interfaces LIBADD contienen la variable \$(DFB_LIBS). Por ejemplo: libdirectfbvideoprovider_xine_la_LIBADD = \$(DFB_LIBS) \$(XINE_LIBS).
- FusionSound? = 1.0.0 (DirectFB) - NOTA: Dentro de la carpeta "interfaces" modificar los archivos *Makefile.am* así que todas las interfaces LIBADD contienen la variable \$(DFB_LIBS). Por ejemplo: libifusionsoundmusicprovider_mad_la_LIBADD = \$(DFB_LIBS) \$(MAD_LIBS)

Cómo compilar los paquetes Ginga-NCL

Antes de empezar la compilación, asegúrese de que las variables de entorno necesarias están presentes y se inicializa correctamente. Los valores iniciales de estas variables se pueden asignar mediante los siguientes comandos:

```
$ export
LD_LIBRARY_PATH=/usr/local/lib/lua/5.1/socket:/usr/local/lib/ginga:
/usr/local
$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

La variable de ambiente PKG_CONFIG_PATH debe estar configurada antes de la compilación de DirectFB, como pre requisito.

telemidia-util-cpp

Pre-requisitos:

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- libpthread

Compilando:

```
$ chmod 755 ./autogen.sh  
$ ./autogen.sh  
$ make  
$ make install
```

telemidia-links-cpp**Pre-requisitos:**

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- directfb = 1.0.1
- libtiff
- libjpeg
- libpng
- libz
- libpthread
- libssl
- libcrypto
- libgssapi_krb5
- libkrb5
- libcom_err
- libk5crypto
- libresolv
- libgpm

Compilando

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh --enable-graphics --with-directfb --enable-
javascript --without-x --wit
$ make
$ make install
```

gingacc-cpp

gingacc-cm

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- expat >= 1.95 (<http://expat.sourceforge.net>)
- telemidia-util-cpp >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser>)
- /telemidia-util-cpp)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-system

Pré-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- libpng

- libjpg
- libmad
- faad2(./configure --with-mp4v2)
- ffmpeg(./configure --enable-shared --enable-gpl --enable-nonfree --enable-pthreads enable-libfaad --enable-postproc)
- xine-lib >= 1.1.17 (./configure --enable-faad --disable-ffmpeg-uncommon-codecs --with-external-ffmpeg)
- MPlayer essential >= 20061022 (<http://www2.mplayerhq.hu>)
- fusionsound >= 1.0.0 (<http://www.directfb.org>)
- directfb >= 1.0.1 (<http://www.directfb.org>)
- directfb-extra >=1.0 (<http://www.directfb.org>)
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-ic

Pre-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- curl >= 7.16

- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-um

Pre-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-ic (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilación

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-player

Pre-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- directfb = 1.0.1
- lua >= 5.1.2
- luasocket >= 2.0
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- telemidia-links-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>) gingacc-system

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-tuner

Pre-requisitos:

- libtool >= 1.3.4
- autoconf >= 2.13
- automake >= 1.4
- video4linux/headers >= 5
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-tsparser

Pre-requisitos:

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-tuner (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-dataprocessing

Pre-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)

- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-tsparser (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-contextmanager

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- lua >= 5.1.2
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-system (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingacc-multidevice

- libtool >= 1.3.4 (hasta 1.5.x)

- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-system (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

ncl30-cpp

ncl30

Pre-requisitos:

- libtool >= 1.3.4 (até 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

ncl30-converter**Pré-requisitos:**

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- xerces-c >= 2.7.0
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingancl-cpp**Pre-requisitos:**

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- ncl30 (from ncl30-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/ncl30-cpp>)

- ncl30-converter (from ncl30-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/ncl30-cpp>)
- gingacc-system (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-ic (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-player (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-contextmanager (from gingacc-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

gingalssm-cpp

Pre-requisitos:

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- telemidia-util-cpp >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/telemidia-util-cpp>)
- gingacc-cm (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- ncl30 (from ncl30-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/ncl30-cpp>)
- ncl30-converter (from ncl30-cpp) >= 0.11.1
(<http://svn.softwarepublico.gov.br/trac/ginga/browser/ncl30-cpp>)
- gingacc-system (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br>)

/trac/ginga/browser/gingacc-cpp)

- gingacc-player (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingancl >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingancl-cpp>)
- gingacc-tuner (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-tsparser (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingacc-dataprocessing (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

ginga-cpp

Pre-requisitos:

- libtool >= 1.3.4 (hasta 1.5.x)
- autoconf >= 2.13
- automake >= 1.4
- gingacc-cm (from gingacc-cpp) >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingacc-cpp>)
- gingancl >= 0.11.1 (<http://svn.softwarepublico.gov.br/trac/ginga/browser/gingancl-cpp>)

Compilando:

```
$ chmod 755 ./autogen.sh
$ ./autogen.sh
$ make
$ make install
```

Utilizando Ginga-NCL versión C++

Para presentar un documento NCL a partir de un archivo con el siguiente comando:

```
$ /usr/local/sbin/ginga --ncl /ruta/para/archivo.ncl
```

Para proporcionar un flujo de transporte para unidifusión UDP compatible con la norma SBTVD-T, edite el archivo `/usr/local/etc/ginga/files/tuner/tuner.ini` coloque la IP del servidor de flujo de transporte. Use el siguiente comando:

```
/usr/local/sbin/ginga
```

Anexo E. Compilación e Instalación de OpenGinga

Este tutorial fue creado tomando como base Ubuntu 10.04 de 32 bits.

Instalando las dependencias con *aptitude*

A mayoría de los archivos necesarios están disponibles en los repositorios de Ubuntu. Para instalar hacer lo siguiente en un terminal:

```
$ sudo aptitude update
$ sudo aptitude install automake subversion git-core build-
essential patch libssl-dev libcppunit-dev autoconf libcurl4-
openssl-dev libreadline6-dev libexpat1-dev libxerces-c2-dev
libmad0-dev libtiff4-dev libkrb5-dev libgpm-dev x11proto-xext-dev
libxext-dev libpng12-dev libjpeg62-dev libfreetype6-dev
libavcodec-dev libavformat-dev libxine-dev libxine1 libxine1-
ffmpeg libdirectfb-extra libtool liblua5.1-0-dev
```

Instalando los codecs de audio y video

Codec-a52

```
$ wget http://liba52.sourceforge.net/files/a52dec-0.7.4.tar.gz
$ tar -xzvf a52dec-0.7.4.tar.gz
$ cd a52dec-0.7.4
$ ./configure
$ make
$ sudo make install
```

Codec-faad

```
$ wget http://downloads.sourceforge.net/faac/faad2-2.7.tar.gz
$ tar -xzvf faad2-2.7.tar.gz
$ cd faad2-2.7
$ ./configure --with-mp4v2
$ make
$ sudo make install
```

Codec-ogg

```
$ wget http://downloads.xiph.org/releases/ogg/libogg-1.2.2.tar.gz
$ tar -xzvf libogg-1.2.2.tar.gz
$ cd libogg-1.2.2
$ ./configure
$ make
$ sudo make install
```

Codec-theora

```
$ wget http://downloads.xiph.org/releases/theora/libtheora-
1.1.1.tar.bz2
$ tar -xjvf libtheora-1.1.1.tar.bz2
$ cd libtheora-1.1.1
$ ./configure
$ make
$ sudo make install
```

Codec-x264

```
$ wget http://hio.googlecode.com/files/x264-snapshot-20100629-
2245.tar.bz2
$ tar -xjvf x264-snapshot-20100629-2245.tar.bz2
$ cd x264-snapshot-20100629-2245
$ ./configure
$ make
$ sudo make install
```

Codec-vpx

```
$ wget http://webm.googlecode.com/files/libvpx-0.9.1.zip
$ unzip libvpx-0.9.1.zip
$ cd libvpx-0.9.1
$ ./configure
$ make
$ sudo make install
```

FFmpeg

```
$ wget http://www.ffmpeg.org/releases/ffmpeg-0.6.1.tar.bz2
$ tar -xjvf ffmpeg-0.6.1.tar.bz2
$ cd ffmpeg-0.6.1/
$ ./configure --enable-shared --enable-gpl --enable-nonfree
--enable-pthreads --enable-libfaad --enable-postproc --enable-
```

```
libtheora --enable-libx264
$ make
$ sudo make install
```

Instalando dependencias de las bibliotecas gráficas

Linux-Fusion

```
$ wget files.openginga.lavid.ufpb.br/deps/linux-fusion.tar.gz
$ tar -xzf linux-fusion.tar.gz
$ cd linux-fusion
$ git checkout af15f7396d96c5d7079ca75703ccb128065a5196
$ wget
http://gingacdn.lavid.ufpb.br/attachments/download/326/fusion_kern
el_version.patch
$ patch -Np0 < fusion_kernel_version.patch
$ make
$ sudo make install
$ sudo depmod -a
$ sudo modprobe fusion
$ sudo chmod 777 /dev/fusion*
```

Para que comando *modprobe* agregue con comandos al arrancar el sistema realizar lo siguiente:

```
$ sudo su
$ echo "fusion" >> /etc/initramfs-tools/modules
$ update-initramfs -u
$ exit
```

DirectFB

```
$ wget http://directfb.org/downloads/Core/DirectFB-1.4/DirectFB-
1.4.5.tar.gz
$ tar -xvzf DirectFB-1.4.5.tar.gz
$ cd DirectFB-1.4.5
$ ./configure --enable-multi --enable-x11 --with-gfxdrivers=none
$ make
$ sudo make install
```

Fusion-Sound

Descargar y descomprimir *FusionSound*

```
$ git clone
git://git.directfb.org/git/directfb/core/FusionSound.git
$ git checkout 6c9f3e777453b0cdbdf3e1db57ae4bfd7dd922d5
$ ./autogen.sh
$ make
$ sudo make install
```

DirectFB-Extra

El archivo *DirectFB-Extra* disponible en los repositorios no está compilado, por lo que el primer paso es el siguiente:

```
$ git clone git://git.directfb.org/git/directfb/extras/DirectFB-
extra.git
$ cd DirectFB-extra
$ git checkout 9e63dda95c580f72f8b54de7cae79ded76823853
$ ./autogen.sh
$ make
$ sudo make install
```

Instalando otras dependencias

Libtool-1.5.26

```
$ wget files.openginga.lavid.ufpb.br/deps/libtool-1.5.26.tar.gz
$ tar -xvzf libtool-1.5.26.tar.gz
$ cd libtool-1.5.26
$ ./configure --prefix=/usr/local/libtool-1.5.26
$ make
$ sudo make install
```

Lua

```
$ wget files.openginga.lavid.ufpb.br/deps/lua-5.1.4.tar.gz
$ tar -xvzf lua-5.1.4.tar.gz
$ cd lua-5.1.4
$ make linux
```

```
$ sudo make install

$ cd ..

$ wget files.openginga.lavid.ufpb.br/deps/luasocket-2.0.2.tar.gz
$ tar -xvzf luasocket-2.0.2.tar.gz
$ cd luasocket-2.0.2
$ make
$ sudo make install

$ sudo ln -s /usr/local/lib/lua/5.1/socket/core.so
/usr/local/lib/lua/5.1/socket/libcore.so
```

Sun J2SE SDK 1.4.2

```
$ wget files.openginga.lavid.ufpb.br/deps/j2sdk1.4.2_19.tar.gz
$ tar -xzvf j2sdk1.4.2_19.tar.gz
$ sudo cp -r j2sdk1.4.2_19 /opt
```

jlibcpp

```
$ svn co -r136 https://jlibcpp.svn.sourceforge.net/svnroot/jlibcpp
jlibcpp
$ cd jlibcpp
$ make
$ sudo make install
```

Flexcm

```
$ wget files.openginga.lavid.ufpb.br/deps/flexcm2.tar.gz
$ tar -xzvf flexcm2.tar.gz
$ cd flexcm/
$ make
$ sudo make install
```

Configurando DirectFB para X11

Recomendamos utilizar *FrameBuffer* como cliente de X11. Esto hace que sea más fácil el depurar el middleware o sus aplicaciones. Para esto hay que crear el archivo `~/directfbrc` con el siguiente comando:


```
$ echo -e "system=x11 \nmode=960x540 \npixelformat=ARGB \nquiet
\nno-debug \nno-trace" >> ~/.directfbrc
```

El campo *system* indica que la salida gráfica de DirectFB será en una ventana de X11. En caso de que quiera trabajar directamente con el *FrameBuffer*, modifique el *system=fbdev*. En el campo *mode*, de X11, se define el tamaño de la ventana creada. Utilice el comando *dfbdump* para verificar que las configuraciones contenidas en el archivo *directfbrc* fueron definidas correctamente por DirectFB.

Compilando o middleware

Arquitetura do middleware

Este proyecto contiene las cabeceras de que todos los componentes de GingaCDN que se deben seguir.

```
$ git clone git://openginga.lavid.ufpb.br/architecture.git
$ cd architecture
$ git checkout stable
$ autoreconf
$ ./configure
$ sudo make install
```

El middleware

Para descargar la versión estable del código haga lo siguiente:

```
$ wget files.openginga.lavid.ufpb.br/openginga.tar.gz
$ tar -xzf openginga.tar.gz
```

Después de esto ingrese al directorio creado de openginga y escriba:

```
$ cd openginga
$ source env.sh
$ sudo -E make
```

Espere a que termine de descomprimir, un mensaje advertirá que esta operación

ha finalizado con éxito, para empezar a usar la implementación haga lo siguiente:

```
$ cd openginga/gingacc  
$ sudo -E ./start.sh
```

Anexo F. Sistema de Televisión Digital Terrestre ISDB-Tb

F.1 ESQUEMA DE MODULACIÓN DEL SISTEMA BRASILEÑO DE TV DIGITAL

La imagen de TV de alta definición (HDTV) codificada puede alcanzar la tasa de 20 Mbps.

Para ser transmitida por aire en la banda de 6Mhz del canal de TV., ella necesita ser modulada con una codificación robusta contra interferencias. La recepción de esa señal en ambientes agresivos, como en centros urbanos, requiere que el proceso de demodulación sea capaz de reconocer y distinguir esas interferencias como el ruido creado por el hombre, el multirecorrido y el efecto Doppler.

El Sistema Brasileño de TV Digital utiliza la modulación BST-OFDM, que consiste en la división de la banda útil del canal en 13 segmentos de 428,5 khz cada uno, los cuales pueden ser agrupados para formar hasta tres distintas camadas en el proceso denominado transmisión jerárquica, en que cada camada puede ser modulada con diferentes programas.

La modulación OFDM ofrece robustez a la distorsión de multirecorrido, una característica de ambientes urbanos con múltiples obstáculos. Esa robustez proviene de la utilización de símbolos de corta duración ocupando banda estrecha, asociada a la banda de guarda. Los parámetros de transmisión pueden ser configurados individualmente para cada segmento, aquí referido como segmento OFDM, formando un canal de composición flexible. Este procedimiento de configuración es designado para la estructura de camada jerárquica.

Una de las características importantes e la modulación OFDM es la posibilidad de operar en el esquema Red de Frecuencia Única (SFN), que permite la repetición de la misma señal sin el cambio de frecuencia. Para adecuar la distancia entre las estaciones SFN y dar robustez al efecto Doppler durante la recepción móvil, fueron establecidos tres modos que consisten en diferentes espaciados entre las frecuencias portadoras. Esos espacios son de 3.968 Hz para el modo 1, 1984 Hz en el modo 2 y 992 Hz en el 3. Con este espaciamiento entre frecuencias en el

modo 1 caben 108 portadoras en cada segmento OFDM, en el modo 2, 216 portadoras y en el modo 3, 432 portadoras.

La existencia de frecuencias pilotos, que funcionan como referencia del canal para el receptor que las utiliza para producir la estimativa de canal y ecualización, garantiza la recuperación de la señal misma en ambientes ruidosos.

F.1.1 Estructura del Sistema

El SBTVD está compuesto por los bloques funcionales mostrados en la figura F-1



Figura F-1: Bloques funcionales del sistema

Encoder: procesa la codificación de video y audio utilizando codificadores H264/AVC HP@ L4.0 para video de servicio fijo y H264/AVC BP@L1.3 en el servicio móvil, para codificación de audio usa el codificador MPEG-4/AAC @L.4 para fijo y el MPEG-4/AAC@L2 para móvil, los cuales proporcionan alta calidad de imagen y sonido y elevada tasa de compresión.

Multiplex: combina en un mismo haz de datos los diferentes transport streams enviados por los codificadores.

Modulador: ejecuta la codificación de canal y la modulación basada en la referencia ARIB STD-B31 V 1.6

Transmisor: convierte la señal de FI de 44MHz generado por el bloc de modulación para la frecuencia del canal de transmisión y amplifica la señal hasta la potencia deseada.

Módulos de recepción: tratan de la funcionalidad del terminal de acceso (Set Top Box), demodulando la señal para el display.

F.1.2 Principales características del SBTVD

Los haces de datos codificados provenientes del multiplexador son sometidos al proceso de codificación de canal. Este consiste en la introducción de algoritmos a los datos para facilitar a los receptores reconocer y corregir los errores provocados durante la transmisión de la señal. Después de este periodo, los haces de bits son mapeados y modulados sobre las diversas frecuencias sub-portadoras que componen el espectro OFDM.

En el modo 1, cada segmento posee 108 portadoras, de las cuales 96 son usadas para la transmisión de datos y 12 frecuencias-piloto. En el modo 2, ese número dobla y en el modo 3 es cuatro veces mayor (ver tabla 1). Una vez seleccionado un determinado modo de transmisión, él debe ser común a todas las camadas.

La figura F-2 ilustra ejemplos de configuración de transmisión, siendo el bloc a la derecha de la figura una configuración jerárquica constituida de camada A, ocupando el segmento central "0" de la banda, la camada B con 7 segmentos y la camada C con 5 segmentos.

Los segmentos de orden impar están localizados al lado izquierdo y los de orden par al lado derecho en relación al centro de la fila. Cada uno de ellos puede ser configurado sin la participación de otros. El número de segmentos agrupados en cada camada jerárquica puede ser seleccionado por el radiodifusor de acuerdo con la intención del servicio que pretende ofrecer. En el sistema brasileño es posible transmitir señales de TV para receptor portátil de banda estrecha usando apenas un segmento OFDM, también llamado one-seg. Este método es denominado recepción parcial y es también considerada una camada jerárquica.

El circuito de transmisión es dividido en tres secciones: codificación de canal, modulador y sección de RF, presentada en la figura F-3

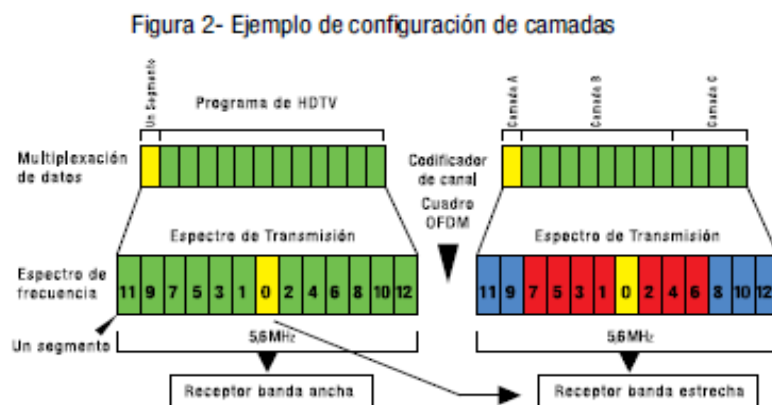


Figura F-2: Ejemplo de configuración de camadas

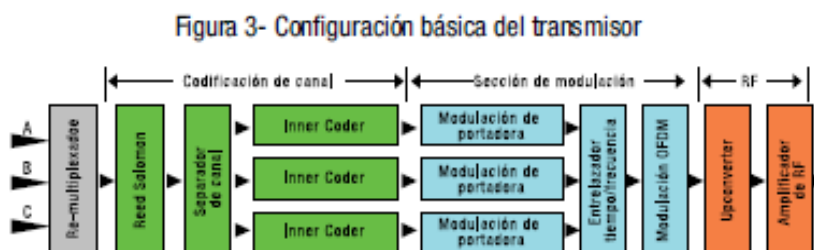


Figura F-3: Configuración básica del transmisor

El módulo re-multiplexador reúne hasta tres haces de transport stream (TS). Los haces son provenientes de distintos codificadores y forman un único haz de datos que será sometido al bloc corrector de errores Redd Solomon acortado siendo adicionados 16 bytes a los 188 bytes iniciales (204,188). Después de este periodo, el TS es dividido nuevamente por el separador de canales, en sus contenidos originales, en paquetes de 204 bytes (TSP), para ser sometido al codificador convolucional(inner code).

Los principales parámetros del SBTVD son mostrados en la tabla 1. La tasa útil de bits de transmisión asume valores diferentes dependiendo del esquema de

modulación, de la tasa de código convolucional y del intervalo de guarda escogidos. La selección de configuración para tasa útil mayor orna el sistema menos robusto.

Sección de Codificación de Canal

El esquema de codificación de canal objetiva introducir algunos algoritmos a la señal, para auxiliar el receptor a reconocer y corregir los errores causados por el canal de transmisión.

La figura 4 muestra los períodos de procesamiento de bits.

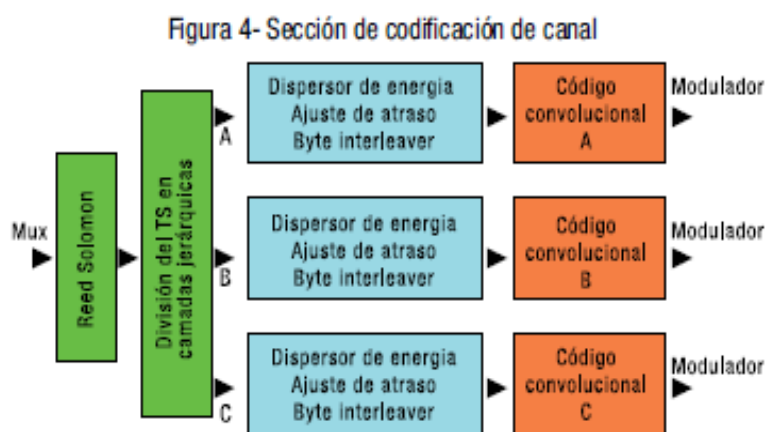


Figura F-4: Selección de codificación de canal

El reed Solomon es un corrector de blocs que, aplicado colectivamente al transport stream total, irá a formar el paquete de datos del canal. En cada símbolo de 188 bytes son adicionados más 16 bytes de paridad. Así, cada símbolo es capaz de corregir hasta 8 bytes errados.

En el caso de la transmisión jerárquica, el transporte stream resultante es nuevamente dividido para formar el conjunto de informaciones de los paquetes originales, en un máximo de tres streams paralelos de procesamiento.

A seguir, el dispositivo dispersor de energía, cuyo objetivo es evitar la repetición de grande secuencia de 1 o 0, es aplicado en cada sección del procesador paralelo usando un circuito PRBS (pseudo Random Bit Sequence).

El ajuste de atraso, asociado al byte interleaving, objetiva la compensación de tiempo para ecualizar el tiempo de transmisión y recepción de todas las capas y es siempre conducido por el lado de la transmisión.

La suma de todos los atrasos, incluyendo el de transmisión y recepción causados por el bit interleaving, es siempre equivalente a la longitud de un cuadro.

El codificador interno es un convolucional activado con código madre de $\frac{1}{2}$ y tiene la longitud de compresión k de 7. En seguida, es efectuada la activación para la tasa de $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, $\frac{5}{6}$ y $\frac{7}{8}$. Ejemplificando: tasa $\frac{3}{4}$ significa que para cada 3 bits de entrada salen 4 bits del codificador.

Los grados de robustez y flexibilidad pueden ser conseguidos especificando diferentes conjuntos de parámetros de transmisión, tales como, el número de segmentos, la tasa de codificación interna y el esquema de modulación para diferentes capas jerárquicas conforme el tipo de servicio que se propone a proveer.

Sección de modulación

Esta sección describe la secuencia de procesamiento de bits suministrados por la sección de codificación de canal para ser modulados. En el proceso de modulación de las portadoras, los bits de la señal de entrada son entrelazados y mapeados por el esquema definido para cada capa jerárquica.

La señal de entrada en el mapeador debe ser de 2 bits por símbolo para modulación en QPSK, mapeado para los ejes I y Q de 4 bits para modulación en 16QAM mapeado para los ejes I y Q y de 6 bits por símbolo para modulación y 64QAM mapeado para los ejes I y Q. Como el número de bits por símbolo para modulación es 64QAM mapeado para los ejes I y Q. Como el número de bits por símbolo aumenta de 2 para 4 y de ahí para 6, la tasa bits aumenta en la misma proporción. Al mismo tiempo, la distancia entre portadoras también disminuye y la configuración queda menos robusta, pero la tasa útil de la señal transmitida aumenta.

Para proceder al mapeo son inseridos en la entrada del mapeador 120 elementos de bits de atraso en el momento de entrelazamiento de bits para la modulación en la Figura 5-Configuración para modulación de las portadoras Figura 4- Sección de codificación de canal QPSK. Para proceder al mapeo en 16 QAM, no es introducido atraso en el primer bit. Pero es introducido atraso de 40 elementos de bits para el segundo bit, atraso de 80 elementos de bits para el tercer bit y 120 elementos de bits para el cuarto bit. Vea en la figura F-5.

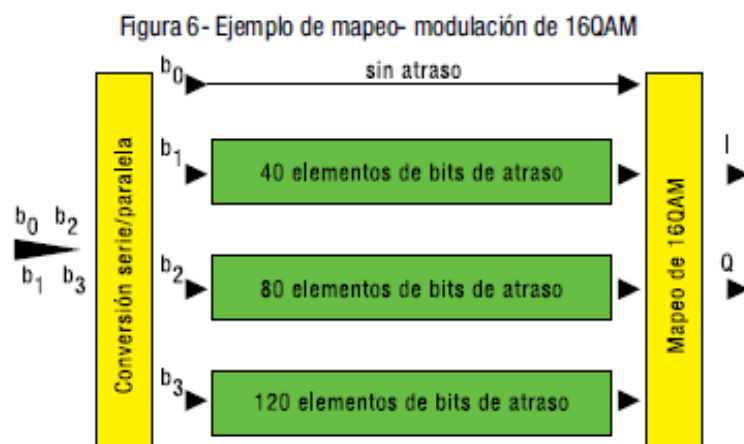


Figura F-5: Ejemplo de mapeo modulación de 16QAM

Existe una correlación entre la tasa de bit transmitida y la robustez de la señal contra los efectos de la interferencia. Entonces, considerando un intervalo de guarda de 1/8 en la modulación QPSK con tasa de C/N de 10dB, hay recepción de excelente calidad. Entre tanto, la tasa de bits transmitida es limitada a 10Mbps. Para la modulación en 64QAM se necesita de C/N de 18dB para garantizar una buena recepción, sin embargo, la tasa de bits transmitida sube para aproximadamente 20Mbps.

Debido a que el nivel de energía de las portadoras, desde que modulados con alto número de estados, es mayor que aquel con menos número de estados, el nivel de la señal de transmisión necesita ser ecualizada para que las potencias medias de las portadoras queden aproximadamente constantes, independientemente del esquema de modulación utilizado. La tabla F-1 muestra los factores de normalización propuestos.

Tabla 2- Tabla del factor de normalización

Modulación	Factor
QPSK	$z/\sqrt{2}$
16QAM	$z/\sqrt{10}$
64QAM	$z/\sqrt{4}$

Tabla F-1: Tabla del factor de normalización

Las señales de diferentes camadas jerárquicas parametradas para diferentes configuraciones necesitan ser combinadas a fin de que sean sometidas en común al proceso matemático de conversión IFFT (Inverse Fast Fourier Transformer).

Las señales procesadas de esta manera son sometidas a time interleaving, en unidades de símbolos de modulación, para asegurar mejor robustez contra interferencia de fading y también pasan por el proceso de frequency interleaving, acción que refuerza el efecto del time interleaving.

En la estructura de cuadro, además, son adicionadas las siguientes señales de frecuencia piloto:

TMCC: Señal que conduce las informaciones de control. El TMCC soporta el receptor en la demodulación y decodificación de varias informaciones, incluyendo identificación de parámetros de transmisión, indicador de cambio de contexto, flag para alarma de emergencia, información de configuración jerárquica actual y parámetros de configuración para la próxima conmutación. El piloto es transmitido en BQPSK suministrando extrema y robusta información de control como el código de sincronismo.

CP: piloto continuo. Sirve como señal de referencia para la sincronización e información para estimativa y ecualización del canal a ser procesado en el receptor.

SP: Piloto dispersado. Es inserido en el segmento a cada 12 portadoras de datos,

dentro de cada fila en la dirección del cuadro OFDM y a cada 4 símbolos en la dirección del símbolo (columnas). El representa 8% de la energía total transmitida.

AC: Piloto auxiliar. Es una señal de extensión que transmite información adicional para control de la señal de modulación.

Tabla 1- Resumen de las características técnicas

	Parámetros	Valores
1	Número de segmentos	13
2	Anchura del segmento	6000kHz /14 = 428,57kHz
3	Anchura de banda útil	(Modo 1): 5,575 kHz (Modo 2): 5,573 kHz (Modo 3): 5,572 kHz
4	Número de portadoras	(Modo 1): 108 por segmento, total 1.405 (Modo 2): 216 por segmento, total 2.809 (Modo 3): 432 por segmento, total 5.617
5	Método de modulación	DQPSK, QPSK, 16QAM , 64QAM
6	Duración de los símbolos activos	(Modo 1): 252µs (Modo 2): 504µs (Modo 3): 1008µs
7	Espaciado entre portadoras	(Modo 1): 428,57kHz/108=3.968Hz (Modo 2): 428,57kHz/216=1.984Hz (Modo 3): 428,57kHz/432= 992Hz
8	Duración del intervalo de guarda	1/4; 1/8; 1/16; 1/32 da duração do símbolo ativo (Modo 1) 63; 31,5; 15,75; 7,875 µs (Modo 2) 126; 63; 31,5; 15,75 µs (Modo 3) 252; 126; 63; 31,5 µs
9	Duración total del símbolo	(Modo 1) 315; 283,5; 267,75; 259,87 µs (Modo 2) 628; 565; 533,5; 517,75 µs (Modo 3) 1260; 1134; 1071; 1039,5 µs
10	Duración del cuadro de transmisión	204 símbolos OFDM
11	Inner code	Código convolucional taxa mãe=1/2; 64 estados Tasa de puncionamiento: 2/3; 3/4; 5/6; 7/8
12	Entrelazamiento inner code	Entrelaçamento intra e inter segmento (frequency interleaving) Entrelaçamento convolucional 0; 380; 760; 1520 símbolos (time interleaving)

Tabla F-2: Resumen de las características técnicas

La señal emergente de la estructura del cuadro OFDM es sometida al proceso de IFFT (Inverse fast Fourier transformer) para generar la señal de FI de 44MHz. Como la señal OFDM es constituida por diversas portadoras ortogonalmente moduladas, cada símbolo es considerado como un elemento de longitud T_u .

Después de la modulación OFDM, es insertado a la señal el intervalo de guarda. Se trata de una extensión cíclica de símbolo OFDM.

El intervalo de guarda permite al receptor eliminar interferencias entre símbolos sucesivos, desde que la dispersión de los tiempos de propagación de todos los multirecorridos envueltos sea menor que el intervalo de guarda. El sistema estandarizó cuatro tiempos de intervalo de guarda: $1/4$, $1/8$, $1/16$ y $1/32$ de la duración del símbolo. No obstante, para la mayoría de las regiones urbanas de Brasil el intervalo de $1/8$ o $1/16$ se mostró suficiente. La tabla F-2 muestra los valores del intervalo de guarda para los modos 1, 2 y 3.

Sección de RF

A la salida de la sección de modulación, la señal de FI de 44MHz es convertida para la frecuencia del canal de transmisión y sometida al amplificador de potencia.

El desvío de frecuencia de la portadora, causado por el error de frecuencia de muestra IFFT a cada fin de anchura de banda, debe ser de 1Hz o menos.

Las frecuencias centrales de los canales digitales deben ser dislocadas de $1/7$ MHz o 142,857kHz en relación al centro del canal, proceso denominado decalaje de frecuencia, conforme lo ilustración F-7 presenta las máscaras crítica, suscrítica y no crítica, las cuales deben ser aplicadas conforme a la clase, potencia y localización de las estaciones transmisoras.

Las estaciones de transmisión digital son clasificadas en clase Especial, clase A, clase B y clase C, cuyos valores de las potencias máximas son presentadas en la tabla 3. En relación a la potencia ERP, para cada clase es tomada como referencia una altura de 150 metros sobre el nivel medio del terreno.

Estas potencias fueron definidas considerándose que el sistema digital deberá replicar las actuales estaciones analógicas suministrando aproximadamente la misma cobertura para la clase equivalente. Esto significa que una potencia media del transmisor digital debe ser aproximadamente 20 veces menor que la potencia de pico del transmisor analógico para la misma clase de transmisión.

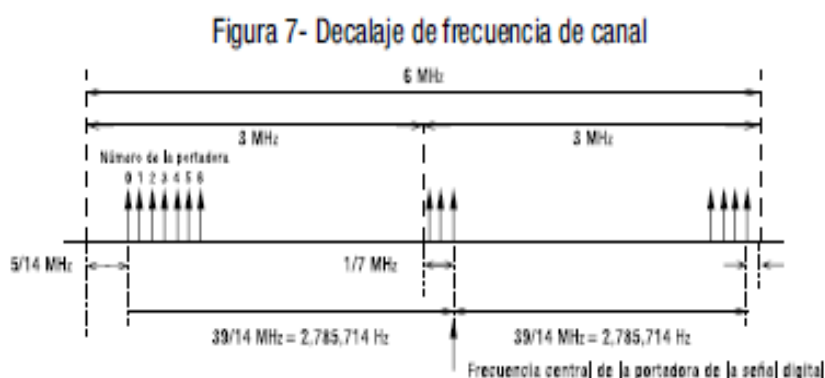


Figura F-6: Decalaje de frecuencia de canal

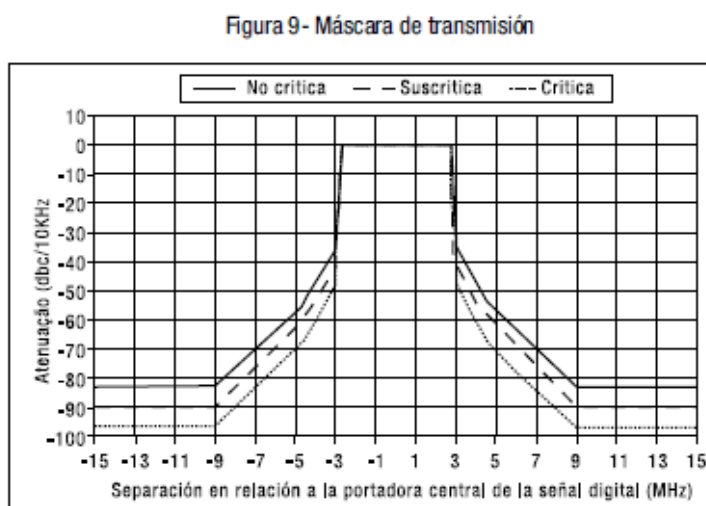


Figura F-7: Máscara de transmisión

F.4 Flexibilidad del SBTVD

Al mismo tiempo en que ofrece robustez a las posibles degradaciones de la señal,

el sistema SBTVD es extremadamente flexible, permitiendo varias configuraciones conforme a la necesidad de cada radiodifusor. La tabla 4 proporciona alguna de las configuraciones posibles para canal de 6MHz, con las tasas útiles de transmisión.

F.5 Conclusiones

El SBTVD es un sistema de transmisión de televisión digital que ofrece robustez a las interferencias y flexibilidad de configuración para atender a las necesidades de cada situación. La configuración puede ser alterada para cada programación de la emisora. Su estructura es basada en el patrón japonés ISDB- T con adición de innovaciones, como el uso de los codificadores H264/AVC y H264/AAC, para vídeo y audio respectivamente, además de las adaptaciones a las condiciones locales.

Glosario

AC: Auxiliary Carrier

ARIB: Association of Radio Industries and Business

BST-OFDM: Band Segmented Transmission OFDM

BQPSK: Binary Quadrature Phase Shift Keying

CP: Continuous Pilot

ERP: Effective Radiate Power

HDTV: High Definition TV

IFFT: Inverse Fast Fourier Transformer

OFDM: Orthogonal frequency Division Modulation

QAM: Quadrature Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

PRBS: Pseudo Random Bit Sequence

SBTVD: Sistema Brasileiro de TV Digital

SDTV: Standard Definition TV

SFN: Single frequency Network

SP: Scattering Pilot

TMCC: Transmission and Multiplexing Configuration Control

TS: Transport Stream

Anexo G. Manual del Usuario

Introducción

La siguiente aplicación concebida para aprovechar las nuevas características de la televisión digital terrestre permite al usuario(televidente) acceder a diversas funcionalidades y aplicaciones desde un menú interactivo, estas aplicaciones tienen como meta proponer una marco de aprovechamiento de las tecnologías de acuerdo a tres áreas específicas: T-Healt, T-Learning y T-Government, haciendo posible un acercamiento virtual a una gran cantidad de personas aprovechando la difusión que un medio de comunicación como lo es la TV representa, lugares donde un acercamiento a las masas es fundamental.

Este manual consta de dos partes:

- **Parte 1:** Presentación de la aplicación DigitalTv
- **Paso 2:** Instrucciones específicas para el correcto funcionamiento de la aplicación.

Paso 1. Presentación de la aplicación DigitalTv

¿Qué es la aplicación DigitalTv?

La aplicación DigitalTv es una herramienta que agrupa los tres principales contextos dentro de lo que se anuncia como el mecanismo para llegar a mucha gente usando las nuevas tecnologías.

¿Qué usuarios podrán acceder a la aplicación?

En términos generales, los usuarios que dispongan de un Set Top Box y una implementación de Ginga, pueden inmediatamente después de su carga verificar que la aplicación esta lista para iniciar. Los usuarios que acceden a la aplicación son todos aquellos que contando con todo el hardware especificado hayan recibido la aplicación con éxito a través del espacio radioeléctrico.

Paso 2. Instrucciones Específicas para un correcto manejo de la aplicación.

Carga de la aplicación

Se puede considerar a la descarga de la aplicación en el Set Top Box, televisor o receptor de televisión digital terrestre como momento inicial. En ese instante un indicador nos avisará cuando la aplicación ha sido recibida correctamente y está lista para su ejecución. Un mensaje igual al de la figura G-1 se mostrará en la esquina superior derecha de la pantalla.



Figura G-1: Aviso de interactividad

La aplicación está lista para ejecutarse y será solo hasta que con el control remoto seleccionado el botón “ROJO” se lance la aplicación, en ese instante el menú principal de aplicación aparecerá por sobre el video de la programación y sin interrumpir su audio. En la figura G-2 es posible apreciar las opciones que son los accesos al despliegue de las demás aplicaciones en el menú. En el contexto del menú tenemos disponibles los siguientes botones(Tabla G-1).

Mientras nos desplazamos por las opciones del menú principal, puede ocurrir dos estados en los que se encuentre alguna opción, estos estados son “Seleccionado” y “No seleccionado”. En la figura G-3(a) es posible apreciar el aspecto de la opción cuando está no seleccionada y en la figura G-3(b) cuando si esta seleccionada.

Aplicación T-Health: Aliméntate Ecuador.

Con esta opción en ejecución, la aplicación Aliméntate Ecuador inicia con los platos distribuidos por la zona de pantalla disponible. No requiere canal de retorno.

Tabla G-1: Tabla de opciones de interacción del menú




	Botón Disponible	Función
	CURSOR DOWN	Desplaza hacia abajo la selección de las opciones del menú
	CURSOR UP	Desplaza hacia arriba la selección de las opciones del menú
	ENTER	Despliega la aplicación de la opción seleccionada.

Figura G-2: Menú principal de la aplicación



Figura G-3: Aspecto de las opciones del menú

En el contexto de la aplicación Aliméntate Ecuador tenemos disponibles los siguientes botones(Tabla G-2).

Tabla G-2: Tabla de opciones para la aplicación Aliméntate Ecuador

Botón Disponible	Función
	CURSOR LEFT Regresa al estado inicial de la aplicación
	BOTONES AMARILLO, AZUL Y VERDE Opciones para escoger el plato requerido.
	BOTÓN ROJO Termina la ejecución de la aplicación y retorna al menú principal

Se muestra las opciones disponibles como se puede observar en la figura G-4. Estas opciones son ilustradas con fotografías de cada plato, la imagen es lo suficiente clara para realizar la mejor elección y la información de la descripción de las funciones de los botones del control remoto permite guiar al usuario de la mejor manera.



Figura G-4: Aliméntate Ecuador, opciones de platos.

Hay tres tipos de platos diferentes de los cuales escogemos alguno de ellos con los botones mencionados anteriormente, en seguida aparecerá el mensaje que corresponde a la elección realizada como se puede apreciar en la figura G-5.

Cada plato tiene su respectiva descripción, y se puede regresar con el botón “CURSOR LEFT” a la vista primera de donde parte la aplicación para realizar una nueva selección si así se desea.



Figura G-5: Mensaje al escoger una opción

Aplicación T-Learning: Historia del Ecuador





La aplicación Historia del Ecuador inicia con la primera pregunta de un cuestionario enfocado en la historia de nuestro país. No requiere canal de retorno.

En el contexto de la aplicación Historia del Ecuador tenemos disponibles los siguientes botones(Tabla G-3).

Se muestra la primera pregunta y a continuación sus respectivas opciones, escoger la que crea que es la correcta usando el control remoto con los botones del teclado numérico, inmediatamente la opción escogida se marca con otro color para distinguir su elección(Figura G-6).

Con los botones de flecha izquierda (“CURSOR LEFT”) y flecha derecha (“CURSOR RIGTH”) puede avanzar o retroceder entre todas las preguntas, la aplicación permite corregir una elección anteriormente realizada las veces que desee, finalmente si quiere ver los aciertos y errores de sus elecciones a las preguntas puede presionar el botón “ENTER”(Figura G-7). Si los resultados son mostrados y alguna pregunta no fue contestada, esta es tomada como incorrecta.

Tabla G-3: Tabla de opciones para la aplicación Aliméntate Ecuador

	Botón Disponible	Función
	CURSOR LEFT CURSOR RIGTH	Avanza y retrocede entre todas las preguntas.
	BOTONES 1, 2, 3, 4	Opciones para escoger el plato requerido.
	BOTÓN ENTER	Visualizar las respuestas, aciertos y errores
	BOTÓN ROJO	Termina la ejecución de la aplicación y retorna al menú principal

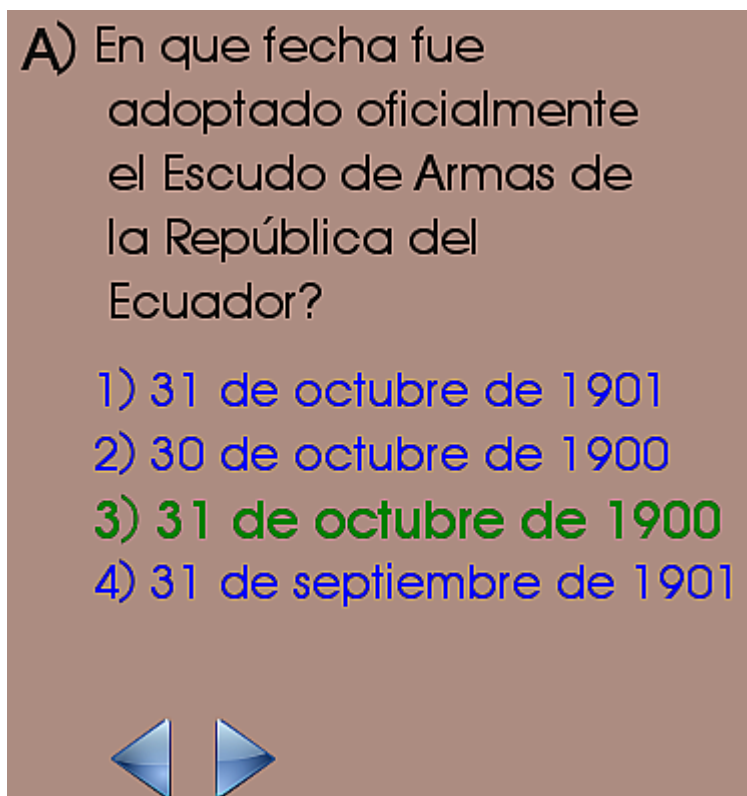


Figura G-6: Historia del Ecuador, pregunta con la opción tres seleccionada



Figura G-7: Historia del Ecuador, mostrando resultados

Aplicación T-Government: Entidades Públicas

La aplicación Entidades Públicas inicia con la petición del ingreso de su cédula de identidad y la posterior consulta de su acta de finiquito. La aplicación requiere canal de retorno.

En el contexto de la aplicación Entidades Públicas tenemos disponibles los siguientes botones(Tabla G-4).

Una marca indica que se trata de un campo de ingreso de datos, con el teclado numérico debe ingresar su número de documento, la aplicación indica con un mensaje cuando los datos ingresados tienen diez dígitos, correspondientes a la longitud de una cédula en nuestro país, así mismo si los datos ingresados es menor a 10 dígitos la aplicación no responderá a la búsqueda cuando se presione el botón “ENTER”, en la figura G-8 se aprecia este detalle.

Tabla G-4: Tabla de opciones para la aplicación Entidades Públicas


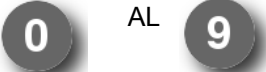


	Botón Disponible	Función
	CURSOR LEFT	Borra el último dígito y establece una nueva búsqueda
	BOTONES 0 al 9	Ingresa los dígitos de su documento de identificación
	BOTÓN ENTER	Indica a la aplicación que empiece la búsqueda
	BOTÓN ROJO	Termina la ejecución de la aplicación y retorna al menú principal



Figura G-8: Entidades Públicas. Ingresando un número de documento

Cuando la búsqueda empieza un mensaje ocupará la pantalla indicando que el proceso esta llevándose acabo, el tiempo de respuesta puede variar dependiendo de su conexión a Internet y de la disponibilidad del servicio por parte del Ministerio de Relaciones Laborales. Dependiendo del caso, la respuesta puede tener tres caminos distintos:

- El usuario no se encuentra registrado en el sistema; que quiere decir que en ninguna ocasión el sistema de Actas de Finiquito fue ocupado para legalizar una

terminación laboral para el usuario consultado.

- El usuario no tiene turnos pendientes; que quiere decir que el usuario si consta registrado en el sistema de Actas de Finiquito pero al momento no tiene turnos por presentarse.
- Información de un turno por atender; en el cual consta de toda la información de la fecha, hora e inspector asignado para atender su pedido de actas de finiquito, figura G-9.

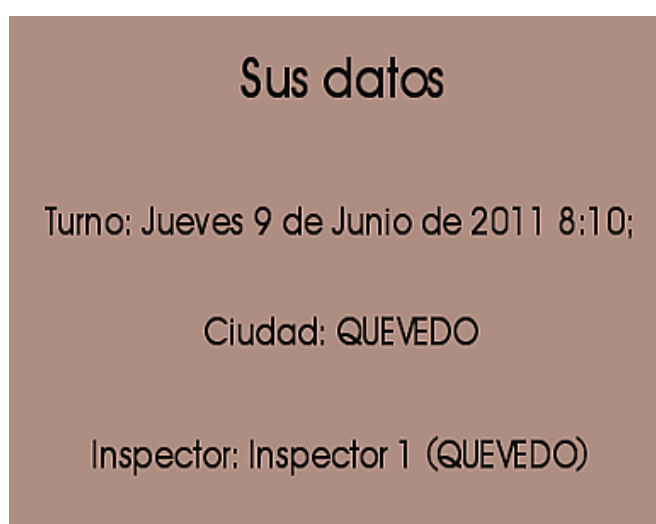


Figura G-9: Información de un turno del Sistema de Actas de Finiquito

Luego de que los resultados se han mostrado puede terminar la aplicación y regresar al menú principal con el botón "ROJO" o con el botón "CURSOR LEFT" prepara el sistema para una nueva búsqueda.

Aplicación Ups

La aplicación Ups cuenta con un segundo menú(figura), que consta con la opción de regresar al menú principal, en el segundo menú su selección desplegará la información de la misión o visión de la Universidad Politécnica Salesiana. La aplicación no requiere canal de retorno.

En el contexto de la aplicación Ups tenemos disponibles los siguientes botones

detallados en la tabla G-5.



Figura G-10: Segundo menú de la aplicación Ups

Tabla G-5: Tabla de opciones para la aplicación Ups

	Botón Disponible	Función
	CURSOR UP y CURSOR DOWN	Permite desplazarse por el segundo menú.
	BOTÓN ENTER	Escoge la opción indicada y ejecuta esa petición
	BOTÓN ROJO	Termina la ejecución de la aplicación y retorna al menú principal

De las opciones disponibles en el menú secundario, la misión(figura G-11) o visión(figura G-12) según sea su elección se mostrará en pantalla.

La aplicación puede terminar y regresar al menú principal con el botón “ROJO”.

Aplicación GamaTv

La aplicación GamaTv interactúa con los servicios de mensajería instantánea de Internet, concretamente la aplicación de microbloggin Twitter, en este caso la cuenta de Twitter de GamaTv está registrada y configurada para disponer de los mensajes más recientes. La aplicación requiere conexión a Internet.

En el contexto de la aplicación GamaTv tenemos disponibles los siguientes botones detallados en la tabla G-6.

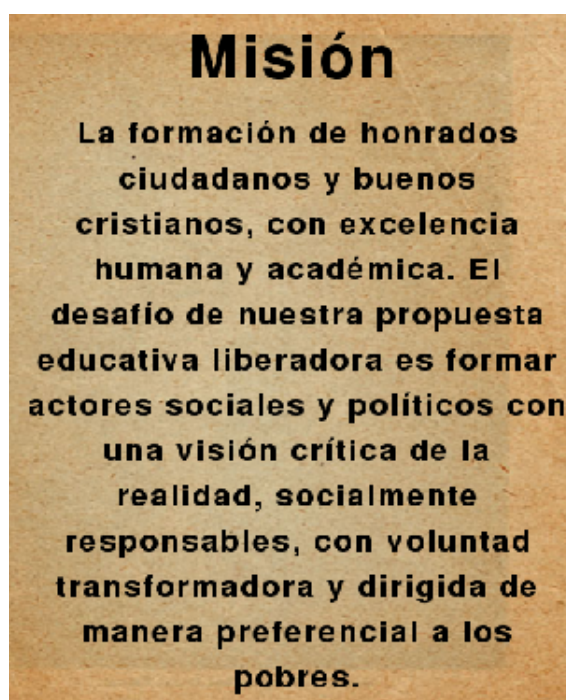


Figura G-11: Opción Misión del segundo menú de la aplicación Ups.

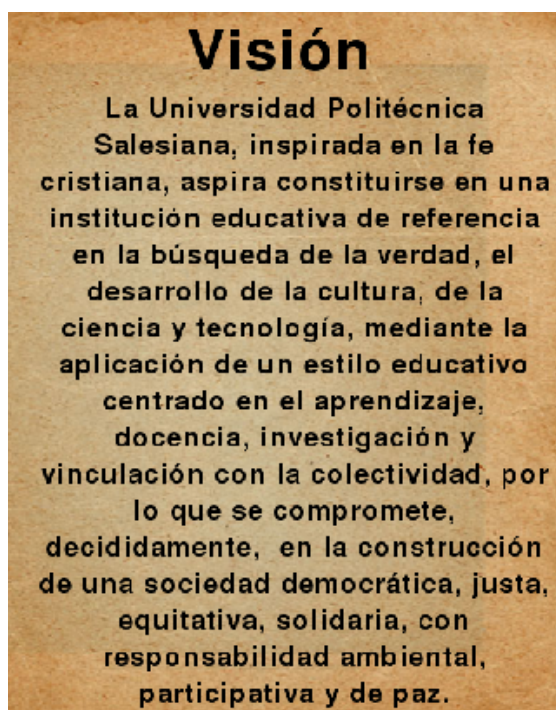




Figura G-12: Opción Visión del segundo menú de la aplicación Ups.

Tabla G-6: Tabla de opciones para la aplicación GamaTv

	Botón Disponible	Función
	CURSOR LEFT CURSOR RIGTH	Avanza y retrocede las publicaciones de twitter
	BOTÓN ROJO	Termina la ejecución de la aplicación y retorna al menú principal

La aplicación en el momento que es ejecutada, empieza la conexión para obtener los mensajes más recientes de la cuenta de twitter de GamaTv, un mensaje advierte de esto al usuario, el tiempo de respuesta varía dependiendo de la velocidad de conexión y la disponibilidad del servicio. Una vez realizada la conexión con éxito el texto de los mensajes aparecerán en pantalla(figura G-13). Con los botones “CURSOR LEFT” o “CURSOR RIGTH” es posible obtener mensajes anteriores o más recientes según sea el caso.

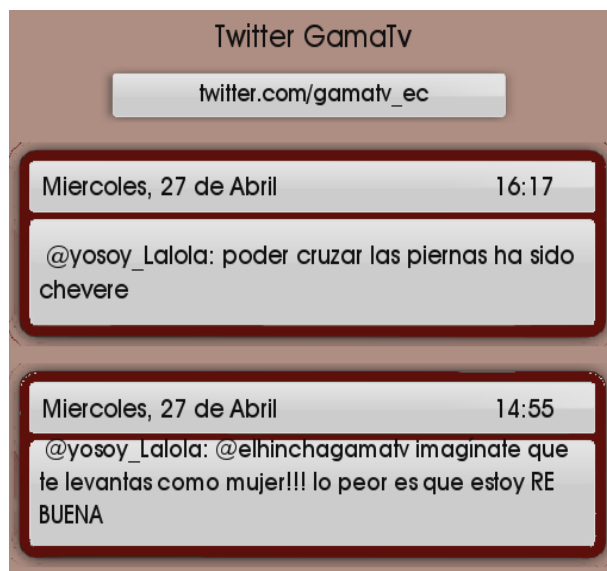


Figura G-13: Aplicación GamaTv, mostrando mensajes de la cuenta de twitter.

La aplicación puede terminar y regresar al menú principal con el botón “ROJO”.

Finalmente para cerrar el menú principal y retornar al video recibido, debemos ubicarnos en la opción “Salir” y presionar el botón “ENTER”.

Anexo H. EITV Payout Professional

La línea de Productos de EITV ya fue aceptada por más de 90 emisoras ISDB-Tb. Los equipos cuentan con redundancia en sus principales componentes (fuentes, discos, etc.) pensados para operar en ambientes de alta disponibilidad. Totalmente compatible con las especificaciones del estándar brasileño SBTVD o ISDB-TB.

El equipo EITV Payout Professional realiza las siguientes funciones:

- Servidor de SI
- Servidor de EPG
- Servidor de Closed Caption
- Servidor de Datos (Ginga)
- Multiplexador
- Remultiplexador



Especificaciones técnicas

SI Server

- SI Multiplexing conforme a norma ISDB-Tb
- Generación de Information Tables: PAT, PMT, NIT, EIT, SDT, TDT, TOT, BIT, SDTT y AIT.
- Configuración de Timezone basado en UTC
- Configuración de todas las tablas que van a ser generadas en el Transport Stream
- Configuración de Virtual Channel
- Configuración de Service ID

EPG SERVER

- EPG Multiplexing conforme a norma ISDB-Tb
- Generación de H-EIT, M-EIT y L-EIT.
- Generación de EIT p/f y EIT scheduling para EPG
- Fecha, hora, duración, título, subtítulo y descripción de programa.
- EIT Descriptors (short event, parental rating, audio component, digital copy control)
- Actualización automática vía archivo XML o configuración vía Web Browser de modo muy intuitivo.

CLOSED CAPTION SERVER

- En conformidad con estándares ABNT NBR 15606-1 y ARIB STD-B24 VOL1 PART 3
- Serial signal input (EIA-608) a través de interfaz RS-232.
- Configuración del PID del output stream de closed caption (CC).
- Soporte de generación de múltiples y simultáneos CC streams (HD, SD, 1SEG, multi-language).
- Generación PTS para stream sincronización A/V
- Exportado en realtime multiplexado en la salida ASI.

DATA SERVER (GINGA/OAD)

- En conformidad con estándares ABNT NBR 15606
- DSM-CC generación de carrusel de objetos.
- Soporte de aplicaciones GINGA-J, GINGA-NCL y GEM.
- OAD: over the air receiver software update;
- Configuración de opción Autostart
- Agenda automático

MULTIPLEXER

- Multiplexación de Transport Stream de acuerdo a estándar ABNT NBR 15603
- Hasta 8 entradas ASI independientes para real-time multiplexing
- Integración con encoders externos a través de las entradas ASI.

- Multiplexación automática de A/V, SI, EPG, closed caption y object carousel.
- Filtro de PIDs, regeneración de tablas TS o BTS en tiempo real.
- Entradas TS o BTS en tiempo real sobre ASI.

RE-MULTIPLEXER

- Re-Multiplexación de Transport Stream de acuerdo a estándar ABNT NBR 15601
- Generación de transport stream organizado en layers (layers A, B, C);
- IIP (ISDB-T Information Packet) packet generation.
- Transmisión de contenido 1-SEG para recepción parcial.
- Generación de señal para HDTV, SDTV y Mobile TV
- BTS real time output sobre interfaz ASI o SPI.

Anexo I. EITV Payout

EITV Payout es una estación completa de transmisión de tv digital interactiva, su uso principalmente se orienta al desarrollo de pruebas de aplicativos.

El equipo EITV Payout Professional integra seis equipos diferentes:

- Servidor de SI
- Servidor de EPG
- Servidor de Closed Caption
- Servidor de Datos (Ginga)
- Multiplexador
- Remultiplexador

